Entwicklung und Evaluierung von verschiedenen Interaktionsparadigmen für das VR Objekt Display

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Informatiker

an der

Fachhochschule für Technik und Wirtschaft Berlin

Fachbereich Wirtschaftswissenschaften II

Studiengang Angewandte Informatik

1. Betreuer: Prof. Dr.-Ing. Thomas Jung

2. Betreuer: Dipl. Ing. Karsten Isakovic

Eingereicht von Stefan Grüttner

Berlin, 18. November 2004

Inhaltsverzeichnis

1 Einführung

1.1 Einleitung

Die adäquate Visualisierung komplexer Daten gewinnt immer mehr an Bedeutung. Dreidimensionale Darstellungen solcher Daten passen besser in das menschliche Vorstellungsvermögen und sind deshalb auf dem Weg zweidimensionale Repräsentationen abzulösen. Eine Interaktion mit diesen Darstellungen sowie die entsprechende Manipulation der Daten spielen hierbei eine ebenso wichtige Rolle. Die Entwicklung von Systemen, die diese Anforderungen erfüllen, ist heute aktueller Forschungsgegenstand mehr denn je.

Der Entwurf und die Entwicklung interaktiver 3-D-Visualisierungssysteme sowie innovativer Mensch-Maschine-Schnittstellen sind Kernkompentenzen des Forschungsbereiches Visualisierungssysteme (ViSTA) am Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik [FIRST]. Die vorliegende Diplomarbeit wurde dort im Rahmen des Projekts *LaserCave* angefertigt. Neben der Entwicklung eines autokalibrierenden Systems auf der Grundlage von Standardprojektionstechnik ist auch der Entwurf und die prototypische Umsetzung eines *VR Objekt Displays* Arbeitspaket des Projekts.

1.2 Aufgabenstellung

Das VR Objekt Display zeichnet sich im Gegensatz zu traditionellen Wandprojektionen durch einen zylindrischen Aufbau aus. Innerhalb des Projektionskörpers kann ein virtuelles Objekt stereoskopisch projiziert werden. Im Rahmen dieser Arbeit sollen Interaktionskonzepte erarbeitet und prototypisch realisiert werden, welche eine betrachterbezogene Darstellung und weitergehende Handlungen mit virtuellen Objekten ermöglichen. Die Ansicht auf das Objekt soll dabei perspektivisch korrekt zu einem Betrachter erscheinen. Das heißt, der Betrachter wird physisch in die Lage versetzt, um das Objekt herumzugehen. Dabei soll das Objekt auf natürliche Art und Weise erfahrbar werden, um ein Gefühl der Präsenz des Objekts innerhalb des Projektionskörpers zu vermitteln. Für die Umsetzung des Interaktionskonzepts sollen verschiedene zur

Verfügung stehende Eingabegeräte in ihrer Eignung getestet und bewertet werden.

1.3 Gliederung der Arbeit

Kapitel 2 vermittelt Wissen, das für das grundlegende Verständnis der Arbeit notwendig ist. So werden wichtige Begriffe wie *Virtuelle Realität* oder *Immersion* erläutert. Weiterhin wird erklärt wie die räumliche Wahrnehmung des Menschen funktioniert und künstlich reproduziert werden kann. Das Kapitel gibt zudem einen Überblick über existierende VR-Systeme und deren Aus- und Eingabegeräte. Des Weiteren werden häufig angewendetet Interaktionsparadigmen beleuchtet.

Kapitel 3 beschreibt zunächst das dem VR Objekt Display zugrunde liegende Software-System. Ferner wird näher auf den Aufbau und die Funktionsweise des VR Objekt Displays sowie im Rahmen der Arbeit eingesetzte Eingabegeräte eingegangen.

Kapitel 4 dokumentiert die Entwicklung von Interaktionskonzepten für das VR Objekt Display und bewertet diese.

Kapitel 5 schildert den Softwareentwicklungsprozess von Anwendungen, die zur Umsetzung der entwickelten Interaktionskonzepte implementiert wurden.

Kapitel 6 gibt ein Resümee der Arbeit und einen Ausblick auf zukünftige Weiterentwicklungen.

2 Grundlagen

2.1 Virtuelle Realität

Schriftsteller wie Aldous Huxley erkannten schon früh das Potential aber auch die Gefahren einer fortschreitenden technologischen Entwicklung. Sein Roman *Schöne neue Welt* [Hux 53] von 1932 versetzt den Leser rund sechshundert Jahre in die Zukunft. Die Charaktere sind Teil einer von Grund auf neu erschaffenen Gesellschaft, die die großen Probleme der Menschheit wie Kriege oder Krankheiten hinter sich gelassen hat und in der alle Menschen gleichermaßen am Wohlstand teilhaben. Preis für dieses scheinbare Paradies ist jedoch die Freiheit und Individualität des Einzelnen. Menschen werden nicht länger auf natürlichem Wege gezeugt, sondern in sogenannten Brut- und Normzentralen geradezu gezüchtet. „Gemeinschaftlichkeit, Einheitlichkeit, Beständigkeit" ist der Leitspruch des Weltstaates. Durch gezielte Konditionierung während der Erziehung und die Befriedigung von Bedürfnissen binnen kürzester Zeit werden die Bürger des Regimes unter Kontrolle gehalten.

Das „Fühlkino" mit „Super-Stereo-Ton" und „Duftorgelbegleitung" ist allseits beliebte Abendunterhaltung. Aldous Huxley erklärt nicht wie die zur Perfektion vollendete Technik funktioniert – was aber auch höchstwahrscheinlich nicht Anliegen seines Romans ist – jedoch beschreibt er wie der technologische Fortschritt die Protagonisten sogar jedes einzelne Haar eines Bärenfells spüren lässt und damit das, was *Virtuelle Realität* heute in den Grundzügen ausmacht.

Aber was genau heißt *Virtuelle Realität*? Dieser Frage soll in dem nun folgenden Abschnitt nachgegangen werden.

Virtuelle Realität (im Folgenden VR genannt) ist das eigentlich semantisch kontroverse Konstrukt zweier Begriffe: *virtuell* und *Realität*. Um VR als solche zu verstehen, sollen diese beiden Begriffe erst einmal getrennt voneinander erläutert werden.

Während das Wort „virtuell" im Kontext der klassischen Physik oder der Optik sei-

nen ganz eigenen Stellenwert besitzt, ist für diese Betrachtung eine Übersetzung aus dem alltägliche Sprachgebrauch mit „scheinbar“ sicherlich ausreichend. Die Erklärung von *Realität* gestaltet sich weitaus schwieriger, da nicht nur in der Philosophie mehrere Definitionsansätze existieren. So geht der Realismus beispielsweise davon aus, dass es eine absolute Realität gibt, die unabhängig von der durch Wahrnehmung im einzelnen Individuum erbauten Realität existiert. Idealisten hingegen sehen Realität als relativ an und betrachten genau diese im Menschen geschaffene Realität als solche. BORMANN gibt eine Definition nach Hans Jörg Sandkühler. „Wirklichkeit ist dann das Synonym für ‚objektive Wirklichkeit‘, oder ‚materielle Wirklichkeit‘: dasjenige, was nicht nur vorgestellt oder gedacht wird, sondern unabhängig von unserem Vorstellen oder Denken an sich besteht“[Bor 94, Seite 24].

Die offensichtliche Unvereinbarkeit der beiden Termini erschwert eine klare Vorstellung vom Begriff *Virtuelle Realität*. Betrachtet man jedoch in der einschlägigen Literatur partiell synonym benutzte Ausdrücke wie *virtuelle Welten*, *virtuelle Umgebung* oder *künstliche Realität*, erschließt sich eher die wirkliche Bedeutung. In Wahrheit ist VR der Versuch, verschiedenste seit den sechziger Jahren laufende Projekte unter einem Oberbegriff zu vereinen. Schon damals entwickelte Systeme erlaubten es über geeignete Ein- und Ausgabemethoden, dreidimensionale Modelle in Echtzeit zu betrachten und mit ihnen zu interagieren.

Einige Definitionen von VR legen besonderen Wert auf die Interaktion sowie auf die Exploration virtueller Welten und betrachten VR vor allem auch als Benutzerschnittstelle. So kann die VR nach BORMANN, auf Grundlage einer von John Walker entwickelten Klassifikation (siehe Tabelle 2-1), auch als die sechste Generation von Computern zu verstehen sein, da sie „den vorerst letzten Stand der Mensch-Computer-Interaktion“ [Bor 94, Seite 19] darstellt.

Ein weiterer Begriff der häufig neben VR auftaucht ist der *Cyberspace*, geprägt durch die Science-Fiction-Trilogie *Neuromancer* von William Gibson [Gib 94]. Dort haben sich die Menschen einen künstlichen Raum geschaffen, der aus dem Abbild sämtli-

Generation	Bedienungsweise	Barriere
1	Schalttafeln, zweckgerichtete Installation	Bedienungsfront
2	Lochkartenstapel, Aufgabenferneingabe	Tresen
3	Fernschreiber-Timesharing	Terminal
4	Menüsysteme	Menühierarchie
5	Bildverarbeitung, Fenster	Bildschirm

Tabelle 2-1: Klassifikation von Computern anhand der Bedienung [Bor 94, Seite 19]

cher auf der Welt gespeicherten Daten besteht. Nicht zu übersehen ist, dass auch das Internet diesen Charakterzug trägt, weswegen der Ausdruck Cyberspace auch gerne in den Medien Verwendung findet. In *Neuromancer* sind die Menschen mit einer direkten Verbindung ihres Nervensystems an die „Matrix“, einem weltweiten Netzwerk, angeschlossen und können so mit allen Sinnen im Cyberspace umherwandeln. Betrachtet man VR als Technik, um als Individuum mit einer künstlichen Welt in Kontakt zu treten, muss man zwischen VR und Cyberspace unterscheiden. Der Cyberspace ist eher „eine Form des Seins“ [Bor 94, Seite 26] und VR in diesem Fall lediglich eine Voraussetzung dafür.

Deutlich wird, dass VR als Begrifflichkeit nicht unumstritten ist. Dies ist vor allem auf „die Unterschiedlichkeit der Interessen der an der Entwicklung beteiligten Personen sowie die Bandbreite möglicher Anwendungen“ [Bor 94, Seite 26] zurückzuführen.

Im Rahmen dieser Diplomarbeit soll VR als Technologie angesehen werden, die einen Menschen in die Lage versetzt, mit einer computergenerierten Welt auf möglichst natürliche Art und Weise in Verbindung zu treten. Das dafür notwendige System umfasst sowohl Hardwarekomponenten, also alle möglichen Ein- beziehungsweise Ausgabegeräte sowie verarbeitende Rechner, als auch Softwarekomponenten, die entsprechende Darstellung und Verhalten (z.B. physikalische Gesetzmäßigkeiten) der Umgebung sicherstellen.

2.1.1 Immersion

Der vom Lateinischen *immersio* abstammende Begriff bedeutet „eintauchen“ und hat in Verbindung mit Virtueller Realität seine ganz eigene Relevanz. Der Hollywood-Film *Matrix* von 1999, für den unter anderem auch William Gibsons *Neuromancer* [Gib 94] als Vorlage gedient haben dürfte, verdeutlicht als Beispiel wohl am besten das Prinzip der Immersion. Die Menschen sind, wie auch in *Neuromancer*, an die „Matrix“ angeschlossen, welche jeden ihrer Sinne künstlich stimuliert. Sie sind Teil einer virtuellen Realität. Nur sind die generierten Sinneseindrücke so überzeugend, dass niemand den Unterschied zur echten Realität zu erkennen vermag. Der Film sowie der Roman sind damit Beispiele für eine perfekte Immersion.

Immersion beschreibt also das Eintauchen in eine künstliche Welt oder vielmehr das Gefühl oder den Grad des „Involviert-Seins“. Heutige VR-Systeme sind natürlich noch weit von der literarischen Science-Fiction entfernt. Der immersive Eindruck, den sie vermitteln können, ist abhängig von ihren teils technischen Eigenschaften. Tabelle 2-2 ist die Abwandlung einer Übersicht nach BORMANN [Bor 94, Seite 84] und verdeutlicht Faktoren, die die Immersion eines Systems beeinflussen beziehungsweise erhöhen.

hohe Bildkomplexität
hohe Bildauflösung
stereoskopisches Sehen
hohe Abdeckung des Gesichtsfeldes
hohe Bildwiederholrate
hohes Maß an Interaktivität (Positionssensoren, etc.)
3D-Ton

Tabelle 2-2: Immersion fördernde Faktoren [Bor 94, Seite 84]

Die Tabelle ist dabei keineswegs vollständig. So fehlen ihr beispielsweise Aspekte, die

kinästhetische Empfindungen erzeugen oder weitere Sinne ansprechen gänzlich. Immersion wächst aber mit der Stimulanz möglichst vieler Sinne. Wie jedoch aus Tabelle 2-2 ersichtlich, wird immersiver Eindruck maßgeblich durch visuelle Gegebenheiten geprägt. Dies ist mit der Tatsache zu begründen, dass das Sehen auch der Sinn ist, der in der menschlichen Wahrnehmung die wichtigste Rolle spielt.

Bezug nehmend auf die Interaktion mit einem System können aber auch Eingabekonzepte die Immersion gravierend beeinflussen. So ist die Eingabe über eine Tastatur heutiger Standard. Ein flüssiges Arbeiten mit dem Zehnfingersystem muss aber langwierig erlernt werden. Spracheingabe hingegen entspricht der natürlichen Ausdrucksform des Menschen, muss nicht erlernt werden und würde damit den Immersionsgrad eines Systems deutlich erhöhen.

2.2 Menschliche Wahrnehmung

Um Sinneseindrücke in einer VR zu immitieren, muss man zunächst wissen, wie Wahrnehmung funktioniert. Glücklicherweise sind heute die physiologischen, psychologischen und biochemischen Zusammenhänge der menschlichen Wahrnehmung weitgehend gut erforscht. Die Erkenntnisse dieser Forschung sind Grundlage dieses Abschnitts, der zeigen will, wie räumliche Wahrnehmung zustande kommt.

2.2.1 Der Sehapparat

Das menschliche Auge hat sich im Laufe der Evolution zu einem perfekten Organ der visuellen Wahrnehmung entwickelt. Seine primäre Funktion besteht darin, elektromagnetische Wellen, speziell das Spektrum des sichtbaren Lichts, in elektrische Impulse umzuwandeln, die vom Gehirn verarbeitet werden können. Den Aufbau des Auges zeigt Abbildung 2-1. Die einzelnen Komponenten erfüllen beim Sehen unterschiedliche Aufgaben. Das Licht tritt zunächst durch den vorderen Teil ins Auge ein. Hier befinden sich Hornhaut, Iris und Linse. Während Hornhaut und Linse lichtdurchlässig sind, ist es die Iris nicht. Sie lässt sich kreisförmig öffnen und schließen und reguliert somit die Intensität des einfallenden Lichts. Hornhaut und Linse des Auges bre-

chen das Licht und bündeln es. Die Brechkraft der Linse ist hierbei variabel, da sie sich durch spezielle Muskeln unterschiedlich stark krümmen lässt. Dieser Vorgang, der ganz automatisch funktioniert, wird als Akkomodation bezeichnet und ist für Nah- und Fernsicht notwendig. Er sorgt dafür, dass der Brennpunkt des Abbildes eines Objekts, das direkt angeschaut wird, genau auf die Fovea fällt. Rein physikalisch entsteht das Bild einer Umgebung also im hinteren Teil des Auges. Dieser Teil ist mit einer Schicht spezieller Zellen bedeckt. Diese Schicht wird Netzhaut oder Retina genannt. Die Zellen sind Photorezeptoren, denen man auf Grund ihrer unterschiedlichen Form die Namen Stäbchen und Zäpfchen gab. Die Netzhaut enthält etwa sechs Millionen Zapfen und 120 Millionen Stäbchen, die nicht nur ganz unterschiedlich auf Farbe und Helligkeit des Lichts reagieren, sondern auch verschieden auf der Retina verteilt sind. Zapfen konzentrieren sich hauptsächlich in der Fovea, die allerdings kein einziges Stäbchen enthält. Die Konzentration der Stäbchen ist hingegen im Umkreis der Fovea am größten und fällt nach außen hin langsam ab. Stäbchen und Zäpfchen besitzen die Fähigkeit, aus Licht elektrische Signale zu erzeugen. Über einen gebündelten Strang von Nervenenden, den Sehnerv, gelangen diese Signale bis in die hinteren Regionen des Gehirns zum visuellen Kortex, wo sie verarbeitet werden und zu einer bewussten visuellen Wahrnehmung führen.

2.2.2 Räumliche Wahrnehmung

Eine dreidimensionale Vorstellung von seiner Umgebung zu haben, ist für jeden Menschen selbstverständlich. Zu verdanken ist sie vor allem unseren zwei nach vorn gerichteten Augen. Diese spezielle Physiologie ist wohl evolutionär zu begründen und lässt sich anhand eines Beispiels aus der Tierwelt verdeutlichen. Die Augen eines Raubtieres, wie zum Beispiel die des Löwen, sind ebenso angeordnet wie beim Menschen, um bei der Jagd die Entfernung zur Beute abschätzen zu können. Potentielle Beutetiere wie die Antilope haben indessen zu den Seiten schauende Augen, die für einen großen Rundumblick sorgen, um Angreifer möglichst früh zu erkennen.

Doch wie kommt der dreidimensionale Eindruck zustande, basiert er doch lediglich

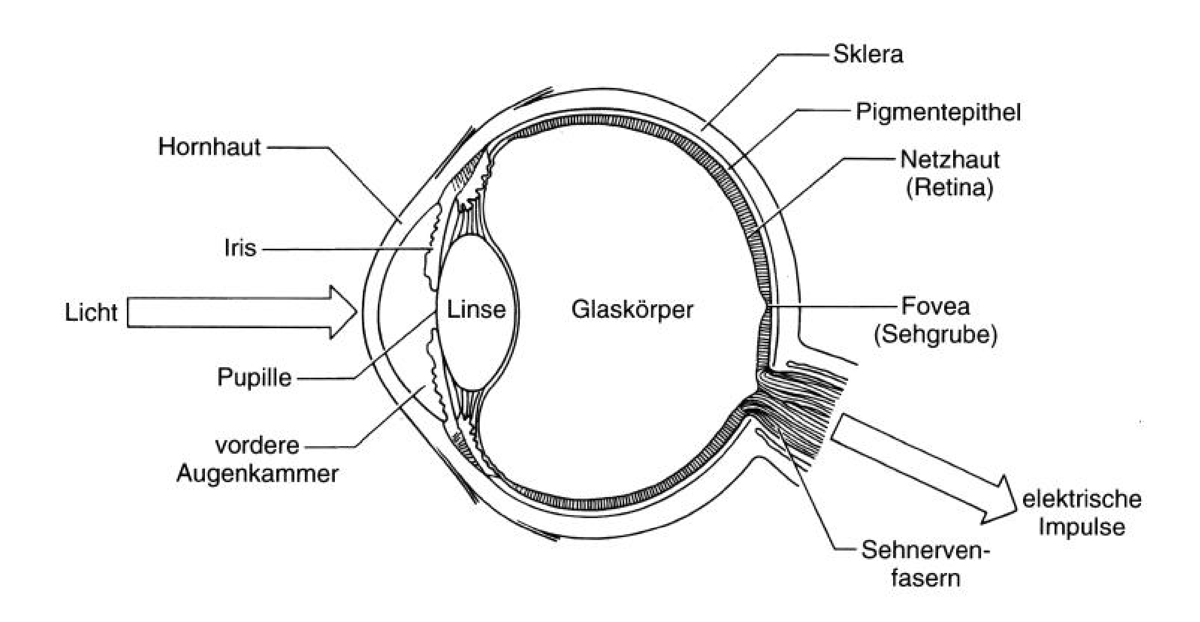


Abbildung 2-1: Querschnitt durch das menschliche Auge [Gold 97, Seite 41]

auf Abbildungen auf der Netzhaut. Offensichtlich besitzt das menschliche Gehirn die faszinierende Fähigkeit aus diesen zweidimensionalen Repräsentationen der Umwelt dreidimensionale Informationen zu generieren, die dann in unsere Wahrnehmung einfließen. In der Wahrnehmungspsychologie existieren mehrere Ansätze, die versuchen diesen Vorgang zu erklären. Einer davon ist die *Theorie der mehrfachen Tiefenkriterien*, die in diesem Abschnitt näher erläutert werden soll.

Die Theorie stützt sich auf eine Reihe von sogenannten Tiefenkriterien, die einen Zusammenhang zwischen den zweidimensionalen Netzhautprojektionen und der wahrgenommenen räumlichen Tiefe herstellen. Nach GOLDSTEIN lassen sich die Tiefenkriterien in vier Gruppen unterteilen [Gold 97, Seite 216ff].

Okulomotorische Tiefenkriterien Betrachtet man ein Objekt in der näheren Umgebung drehen die Augenmuskeln die Augen nach innen, um die Blickachsen beider Augen im anvisierten Punkt (Fixationspunkt) zum Schnitt zu führen. Dieser Vorgang wird als *Konvergenz* bezeichnet. Die Sehachsen schließen dabei den sogenannten Konvergenzwinkel ein. Dieser ist umso größer, je näher sich ein Objekt zum Betrach-

ter befindet und ist somit ein Maß für räumliche Tiefe. Sicherlich berechnet das Gehirn nicht den Konvergenzwinkel, er soll nur eine bessere Vorstellung vom Sachverhalt schaffen. Die Stellung der Augenmuskeln wird dem Gehirn aber bestimmt ebenso viel Aufschluss bieten.

Der Vorgang der *Akkomodation* sorgt dafür, dass das Bild des fixierten Objekts auf der Netzhaut fokussiert wird. Ziliarmuskeln krümmen die flexible Linse, sodass das einfallende Licht entsprechend gebeugt wird und scharf auf der Netzhaut erscheint.

Konvergenz und Akkomodation sind allerdings nur vom Nahpunkt[1] bis zu einer Distanz von maximal drei Metern aussagekräftig.

Monokulare Tiefenkriterien Wie der Begriff bereits vermuten lässt, sind monokulare Tiefenkriterien Anhaltspunkte für räumliche Tiefe, auf die man auch schließen kann, wenn nur ein Auge zum Einsatz kommt.

Bei der *Verdeckung von Objekten* wird die logische Schlussfolgerung gezogen, dass sich ein Objekt im Sichtfeld vor einem anderen befinden muss, wenn es dieses teilweise verdeckt. Diese Implikation beruht auch auf Erfahrungen, die ein Mensch im Laufe seines Lebens macht und ist ein Beispiel dafür, dass auch das Gedächtnis erheblichen Einfluss auf unsere Wahrnehmung hat. Verdeckung kann aber nur relative Aussagen treffen und gibt keinen Aufschluss über absolute Entfernungen.

Das Kriterium der *relativen Größe im Blickfeld* beschreibt das Phänomen, dass in etwa gleich geartete Objekte, die sich im direkten Vergleich in ihrer auf der Netzhaut abgebildeten Größe unterscheiden, als verschieden weit entfernt wahrgenommen werden. Das kleinere der beiden Objekte wird hierbei meist als das fernere identifiziert.

Die *relative Höhe* eines Gegenstandes *im Blickfeld* kann ebenfalls Indiz für räumliche Tiefe sein. Haben Objekte den gleichen Sehwinkel, besitzen also die gleiche projizierte Größe, erscheinen sie weiter weg, sobald sie sich höher im Blickfeld befinden

[1] Als Nahpunkt wird der minimale Abstand bezeichnet, in dem das Auge die Fähigkeit besitzt, ein Objekt scharfzustellen

als andere. Dieser Zusammenhang hat aber nur solange Bestand wie sich die Objekte unterhalb der Horizontlinie befinden. Bei Objekten oberhalb des Horizonts beschreibt sich der Sachverhalt genau umgekehrt.

Staub, Ruß, Nebel und andere Kleinstpartikel in unserer Luft streuen das Licht und lassen ein Objekt zunehmend unscharf erscheinen, je weiter es vom Betrachter entfernt ist. Die sogenannte *atmosphärische Perspektive* ist dabei stark abhängig von der jeweiligen Wetterlage und dem Ort auf der Welt, an dem man sich befindet.

Ein weiteres Tiefenkriterium, das sich auch auf Erfahrung zurückführen lässt, ist die *gewohnte Größe von Gegenständen*. Erblickt man einen alltäglichen Gegenstand wie beispielsweise einen Apfel, kann man relativ gut abschätzen in welcher Entfernung er sich befindet.

Ebenfalls ein Anzeichen für Dreidimensionalität ist die *Perspektive*. Sie lässt sich sehr gut nachvollziehen, wenn man sich vorstellt auf einer Eisenbahnschiene zu stehen, die schnurgerade verläuft. Die in Wirklichkeit parallel verlaufenden Schienen scheinen zu konvergieren und sich in einem Punkt am Horizont, dem Fluchtpunkt, zu treffen. Perspektive wird auch häufig in der Kunst angewendet, um in einem Bild räumlichen Eindruck zu erzeugen.

Das letzte aus der Reihe der Monokularen Tiefenkriterien ist der *Texturgradient*. Ein gutes Beispiel, das hier angeführt werden soll, ist das Schachbrett. Betrachtet man es von einer Seite aus einer etwas gehobeneren Position, scheint sich das perfekte Muster der eigentlich gleichgroßen Quadrate mit zunehmender Entfernung zu verdichten. Dies geschieht erfahrungsgemäß bei allen regelmäßigen Strukturen und kann bei Oberflächen helfen, Entfernungen relativ gut einzuordnen.

Bewegungsinduzierte Tiefenkriterien Die Änderung des Standpunkts kann ganz enscheidend zur räumlichen Wahrnehmung einer Umgebung beitragen. Besonders gut deutlich wird dies bei einer Zugfahrt, bei der man aus dem Fenster blickt. Durchfährt man eine Landschaft stellt man schnell fest, dass Vegetation in unmittelbarer Entferung sehr schnell am Gesichtsfeld vorbeizieht, weit entfernte hingegen nur lang-

sam.

Die sogenannte *Bewegungsparalaxe* ist als Winkeländerung definiert, die infolge einer parallel zu einem Objekt vollzogenen Translation resultiert. Anhand Abbildung 2-2 lässt sich die Bewegungsparalaxe gut veranschaulichen. In diesem Szenario befinden sich die Objekte A und B in unterschiedlicher Entfernung zum Auge. Der Betrachter bewegt sich nun von Position 1 zu Position 2. Die Projektionen der beiden Objekte auf der Netzhaut überstreichen dabei unterschiedlich große Winkel zwischen A_1 und A_2 beziehungsweise B_1 und B_2. Dies entspricht im Falle von Objekt A einer schnelleren Durchquerung des Blickfeldes im Gegensatz zu Objekt B. Im Hinblick auf das oben angeführte Beispiel lassen sich die scheinbar unterschiedlichen Geschwindigkeiten der Objekte mit der Tatsache begründen, dass ihre Abbildungen in einem konstanten Zeitraum verschieden große Strecken auf der Retina zurücklegen.

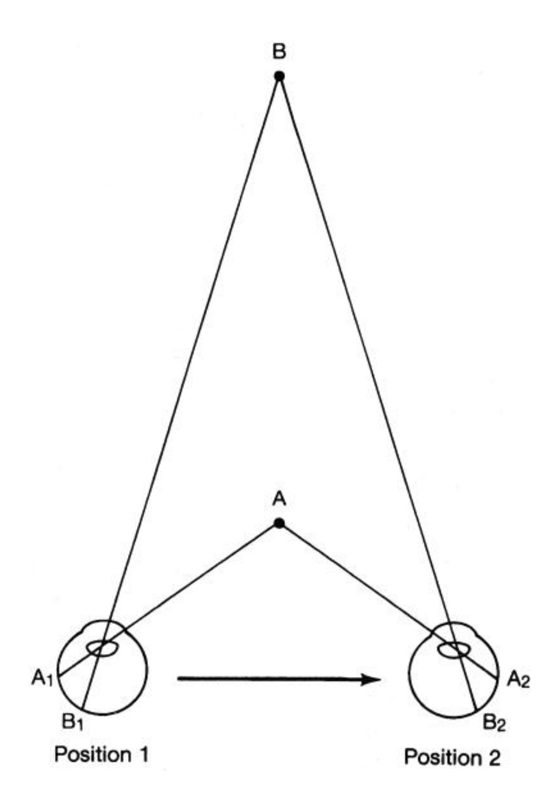


Abbildung 2-2: Bewegungsparalaxe [Gold 97, Seite 223]

Ein eng mit der Bewegungsparalaxe zusammenhängender Effekt zeigt sich auch bei hintereinander liegenden Flächen. Verändert der Betrachter seinen Standpunkt wird die hintere Fläche entweder zunehmend auf- oder zugedeckt, abhängig von der Richtung, in die man sich bewegt.

Querdisparation Abgesehen von der Konvergenz sind alle bisher angeführten Anhaltspunkte für räumliche Tiefe auch nur mit einem Auge nachvollziehbar. Ein großer Vorteil für die räumliche Wahrnehmung ergibt sich aber aus dem Sehen mit zwei Augen. Die *Querdisparation* wird daher auch als *binokulares Tiefenkriterium* bezeichnet. Der Augenabstand lässt in jedem der beiden Augen ein leicht differierendes Abbild der Umwelt auf Grund zweier unterschiedlicher Blickwinkel entstehen. Durch Konvergenz und Akkomodation wird der Fixationspunkt auf die Foveae abgebildet, die sogenannte *korrespondierende Netzhautpunkte* darstellen. Das heißt, einem Punkt der einen Netzhaut ist physisch genau ein Punkt auf der anderen zugeordnet (siehe Abbildung 2-3).

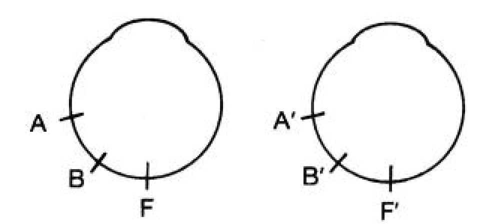


Abbildung 2-3: Korrespondierende Netzhautpunkte [Gold 97, Seite 226]

Es werden aber auch andere Punkte der Szene auf korrespondierenden Netzhautpunkten abgebildet. Diese liegen zusammen mit dem Fixationspunkt auf einer imaginären Linie, dem *Horopter*, da der durch beide Abbildungen eingeschlossene Winkel genau dem Konvergenzwinkel entspricht (siehe Abbildung 2-4).

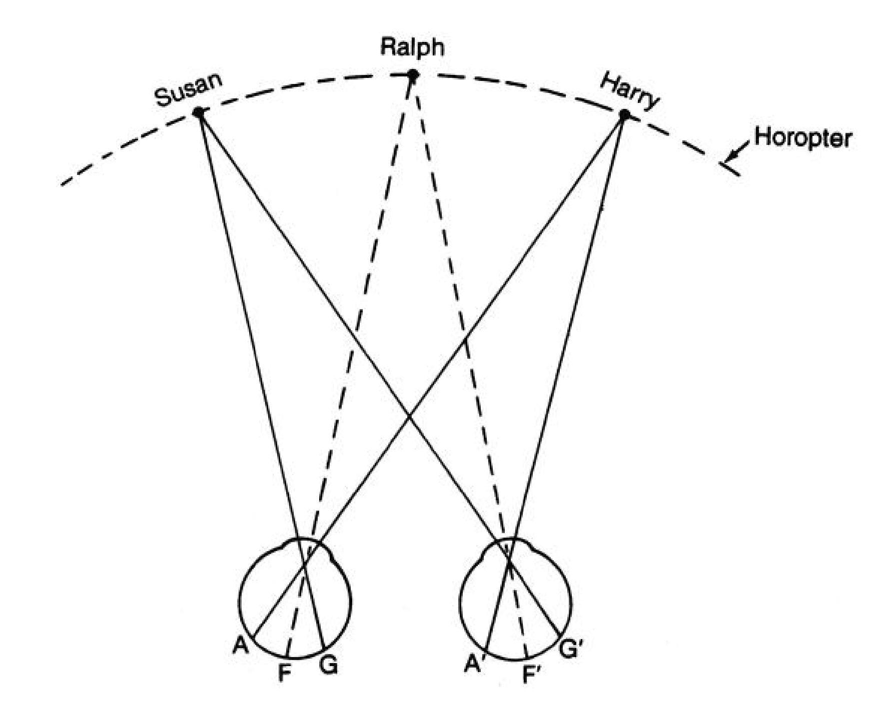


Abbildung 2-4: Horopter [Gold 97, Seite 227]

Das Gehirn ist darüber hinaus aber auch in der Lage in beiden Projektionen diejenigen Stellen zu identifizieren, die in der realen Szene ein und demselben Punkt entsprechen, aber nicht auf dem Horopter liegen. Dementsprechend fallen die Abbildungen dieser Punkte auch nicht auf korrespondierende Netzhautpunkte. Zwischen einem Punktabbild und dem aus dem anderen Auge entstehenden korrespondierenden Netzhautpunkt entsteht der *Querdisparationswinkel*, der als eine hochgradige Quelle für Tiefeninformationen gilt. Wie aus Abbildung 2-5 ersichtlich, fallen die Abbildungen eines Punktes, der vor dem Horopter liegt, auf die jeweiligen Außenbereiche der Netzhäute. Man spricht hierbei von gekreuzter Querdisparation. Bei der ungekreuzten Querdisparation liegen die Punkte hingegen außerhalb des Horopters und die Projektionen liegen zur Nase hin innen auf der Netzhaut. Querdisparation liefert ein starkes Kriterium für räumliche Tiefe und wirkt bis zu Entfernungen von mehreren hundert Metern. Sie gilt als entscheidend für stereoskopisches Sehen.

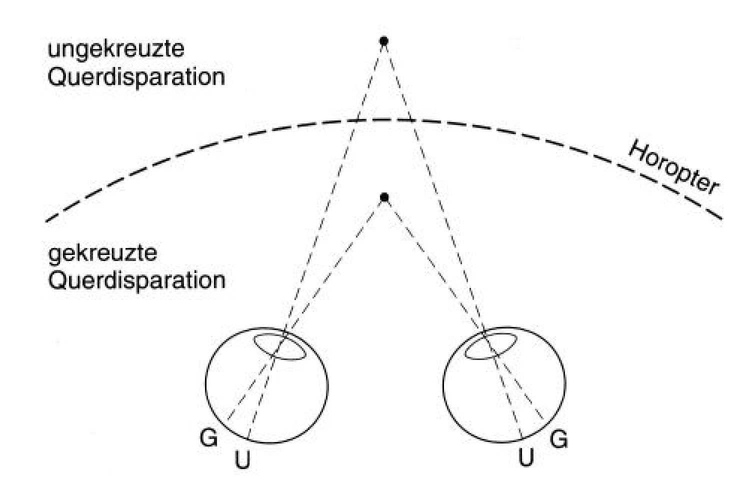


Abbildung 2-5: Gekreuzte bzw. ungekreuzte Querdisparation [Gold 97, Seite 227]

2.2.3 Künstlich induzierte Tiefenwahrnehmung

Die heutige 3D-Computergrafik ist in der Lage, auf Basis von 3D-Modellen Darstellungen zu generieren, die allen monokularen Tiefenkriterien entsprechen. Populäre Programmierschnittstellen wie OpenGL oder Direct3D stellen hierfür einen reichhaltigen Funktionsumfang bereit. Die so entstehenden Bilder können an Bildschirmen angezeigt oder mithilfe von Projektoren auf Leinwände projiziert werden.

Ausreichend dimensionierte Darstellungsflächen, die einen Großteil des Sichtfeldes ausfüllen, sind in Lage auch ohne Stereoskopie einen hohen Grad an Immersion zu erzeugen. Dies ist vor allem auf Objekte zurückzuführen, die auf Grund ihrer scheinbaren Entfernung eine große Wirkung monokularer wie bewegungsinduzierter Tiefenkriterien entfalten.

Um einen annähernd echten räumlichen Eindruck näher liegender Objekte zu erzeugen, muss man Stereopsis schaffen. Hierfür bedient man sich dem Prinzip der Querdisparation. Anstatt nur ein Bild der Szene zu erzeugen, erzeugt man zwei Bilder aus leicht unterschiedlichen Kamerapositionen beziehungsweise -winkeln, um sie je einem Auge gleichzeitig darbieten zu können. Die Augen akkomodieren auf die Entfernung zur Bildebene, also zum Bildschirm oder zur Leinwand, was dem Horopter

entspricht und erfassen durch Konvergenz gemeinsame Bildpunkte. Punkte oder Objekte die gekreuzte Querdisparation erzeugen, scheinen nun vor der Bildebene zu schweben, Objekte mit ungekreuzter Querdisparation hinter ihr (siehe Abbildung 2-5).

Bei diesem Ansatz kann es geschehen, dass die beiden okulomotorischen Reflexe in keinem natürlichen Verhältnis zueinander stehen. Eine solch typische Situation zeigt Abbildung 2-6. Die durch Konvergenz suggerierte scheinbare Entfernung zum Objekt steht im Widerspruch zur tatsächlichen Entfernung zur Bildebene. Des Weiteren können unverhältnismäßig eingestellte Kamerawinkel zu unnatürlichen Augenstellungen wie Divergenz führen. Ein Betrachten solcher Bilder kann daher oft von großer Anstrengung und Unbehagen begleitet sein.

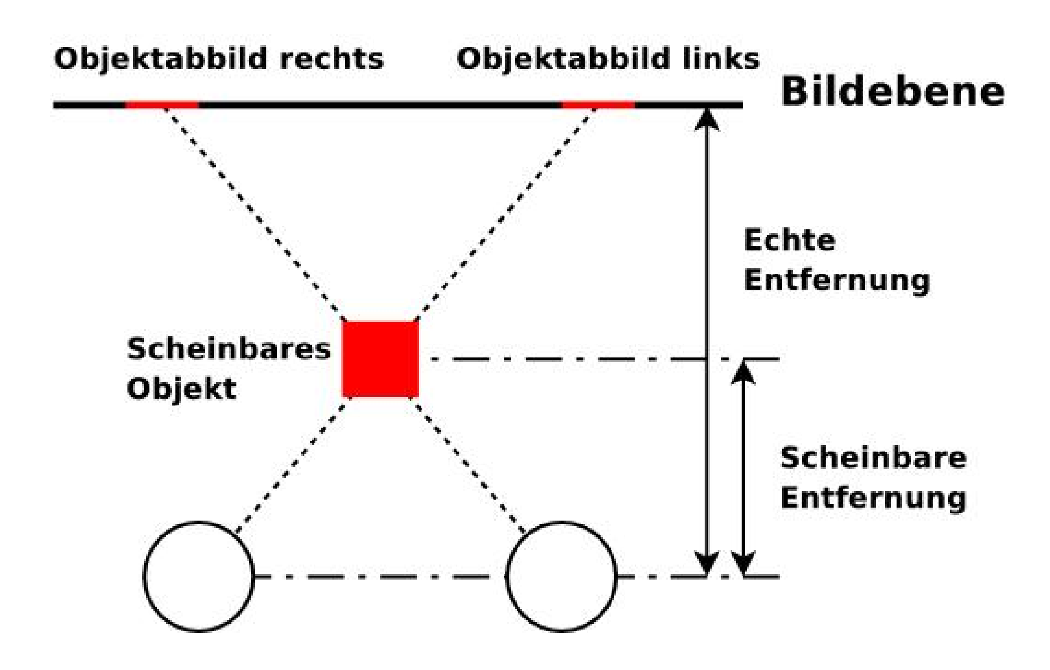


Abbildung 2-6: Konflikt Konvergenz-Akkomodation

Ein weiteres grundsätzliches Problem, das zudem gelöst werden muss, ist die Trennung der beiden Bilder, sodass jedes Auge nur das zugeordnete Bild zu sehen bekommt. Hierzu kommen verschiedene *Bildtrennungsverfahren* zum Einsatz.

Aktive Bildtrennung An Bildschirmen arbeitet man mit einem aktiven Bildtrennungsverfahren, auch Zeitmultiplex-Verfahren genannt, das man auch aus 3D-Kinos kennt. Der Benutzer trägt eine sogenannte Shutterbrille, deren Gläser elektronisch abdunkelbar sind. Das Behindern der Sicht auf den Bildschirm geschieht abwechselnd. Wird einem Auge die Sicht auf den Bildschirm verwehrt, wird dem anderen ein entsprechendes Bild der Szene dargeboten. Die hierfür notwendige Synchronisation zwischen Bildschirm und Brille ist meistens kabellos über Infrarot realisiert. Die Wechselfrequenz entspricht der Hälfte der Darstellungsfrequenz. Letztere ist, bedingt durch die meist verwendete LCD-Shuttertechnik, nach oben hin auf etwa 120Hz begrenzt. Als Minimum wird eine Darstellungsfrequenz von 60Hz angesehen, da das bei niedrigeren Frequenzen infolge des Abdunkelns enstehende Flackern als überaus störend empfunden wird. Bei geeigneter Wechselfrequenz ist das Gehirn nicht mehr in der Lage die Einzelbilder zu trennen, die daraufhin zum erwünschten dreidimensionalen Eindruck verschmelzen.

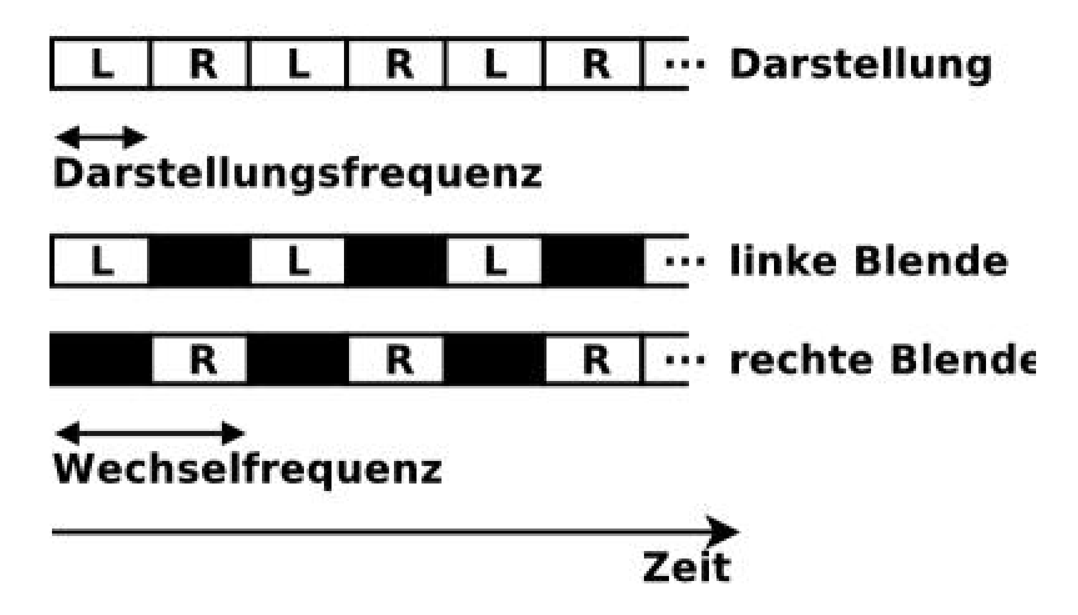


Abbildung 2-7: Aktive Bildtrennung (Zeitmultiplexing)



Abbildung 2-8: LCD-Shutterbrille CrystalEyes 3 von StereoGraphics

Passive Bildtrennung Passive Bildtrennungsmechanismen basieren auf der Filterung des Lichts. Bekannt sein dürfte das anaglyphe Filterungsverfahren. Hierbei werden die beiden Bilder in zwei primären Farbtönen dargeboten, zum Beispiel rot und blau. Eine getragene Brille mit entsprechenden Farbfiltern als Gläser trennt dann die beiden Bilder voneinander. Der Vorteil dieses Verfahrens liegt darin, dass solche Bilder auch abgedruckt oder am Bildschirm ohne zusätzliche Technik dargestellt werden können. Der große Nachteil besteht im Verlust der Farbinformationen des Bildes.

Andere passive Verfahren sind jedoch in der Lage, die Farbinformationen der Bilder aufrecht zu erhalten. Eines basiert auf der physikalischen Annahme, dass sich Licht in Form transversaler Wellen ausbreitet. Angesichts dessen ist es möglich, Licht zu polarisieren. Transversalwellen können als Überlagerung horizontal und vertikal schwingender Wellen betrachtet werden. Mit entsprechend orientierten Polarisationsfiltern besteht die Möglichkeit, einen der beiden Überlagerungsanteile zu eliminieren. Es entsteht linear polarisiertes Licht. Des Weiteren ist die Phasendifferenz der überlagerten Schwingungsanteile Ausschlag gebend dafür, ob eine Transversalwelle links- oder rechtsdrehend ist. Hieraus ergibt sich eine weitere Form der Polarisation. Durch zwei hintereinander liegende und leicht verdrehte lineare Filter kann das Licht zirkular

polarisiert werden. Je nach dem, ob sie gegen den oder mit dem Uhrzeigersinn zueinander verdreht sind, passieren nur links- beziehungsweise rechtsdrehende Lichtwellen die beiden Filter. Im Sinne der Bildtrennung wird das Licht zweier Projektoren zunächst mithilfe entsprechend angeordneter Polarisationsfilter entgegengesetzt (linear oder zirkular) polarisiert. Der Betrachter trägt eine Polarisationsbrille, deren Gläser dieselbe Filterkonfiguration wie die Projektoren besitzen. Auf diese Weise sieht jedes Auge nur sein entsprechendes Bild. Lineare Polarisation hat den Nachteil, dass eine seitliche Neigung des Kopfes den Effekt der Bildtrennung zerstört. Daher zieht man ihr zirkulare Polarisation vor.

Ein relativ neuer Bildtrennungsmechanismus, das sogenannte Wellenlängenmultiplex-Verfahren [INFI], wurde von der Infitec GmbH entwickelt. Infitec steht für Interferenzfiltertechnik. Der Farbwahrnehmung entsprechend zerlegen spezielle Filter die Lichtspektrenanteile der Grundfarben rot, grün und blau in Wellenlängen-Tripel. In Abhängigkeit der Anzahl der zu trennenden Bilder können prinzipiell entsprechend viele Wellenlängen-Tripel erzeugt werden. Aus der Eliminierung von Wellenlängenanteilen der einzelnen Grundfarben ergibt sich jedoch ein Nachteil. Die Farbrezeptoren der Netzhaut reagieren unterschiedlich auf die leicht differierenden Wellenlängen, was sich in einer wahrgenommenen Farbabweichung ins Rötliche beziehungsweise Grünliche der jeweiligen Bilder äußert. Um ungeachtet dessen ein farbtreues Ergebnis zu erzielen, muss der Farbraum der Bilder vor der Filterung adäquat transformiert werden. Infitec stellt hierzu spezielle Hardware zur Verfügung, die zwischen Grafikkarte und Projektor geschalten werden kann. Ein Farbabgleich kann aber auch schon im Rechner mithilfe einer softwareseitigen Lösung bewirkt werden. Der Benutzer muss für einen stereoskopischen Eindruck wieder eine Brille mit entsprechenden Filtern tragen.



Abbildung 2-9: Polarisationsbrille



Abbildung 2-10: Infitec Brille

Einen rein mechanischen Ansatz zur Bildtrennung verfolgt das Head Mounted Display (HMD). Hier werden vor jedes Auge LC-Bildschirme in Miniaturausführungen plaziert. Diese sind in an Helme erinnernde Geräte integriert, die vom Benutzer auf dem Kopf getragen werden (siehe Abschnitt 2.3.1.1).

2.3 VR Systeme

Zur näheren Einführung in die Thematik soll an dieser Stelle zunächst ein historischer Abriss erfolgen.

Seit Erfindung des Computers hat die Entwicklung verschiedenste Varianten der Mensch-Computer-Kommunikation hervorgebracht. Mussten erste Rechner noch mit Lochkarten – für deren Erstellung speziell ausgebildetes Fachpersonal notwendig war – gefüttert werden, haben sich im Laufe der Zeit sehr viel einfachere Ein-/Ausgabegeräte etabliert. So ist heute der Umgang mit Tastatur und Maus zur Bedienung des Computers, verbunden mit einer visuellen Rückkopplung der Eingabe am Bildschirm, selbstverständlich geworden. Hierfür mussten allerdings auch erst Metaphern wie die des Desktops eingeführt werden. Auf einer grafischen Oberfläche werden Anwendungen mithilfe von Symbolen und Fenstern repräsentiert und können mit einem duch die Maus beweglichen Zeiger angewählt werden. Die Navigation wird zudem mit hierarchisch aufgebauten Menüs unterstützt. Die Metapher wird daher auch als WIMP (Windows, Icons, Menus and Pointers) Schnittstelle bezeichnet [Bow 04, Seite 91].

Eine alphanumerische Eingabe erfolgt mit der Tastatur.

Als erste brachte die Firma Apple ein eigenes derartig aufgebautes Betriebssystem zur Marktreife. Andere Hersteller wie Microsoft folgten mit ähnlichen Umsetzungen und ermöglichten von da an erst einer breiten Masse den Umgang mit einem Computer. Neben Maus und Tastatur haben sich heute auch speziellere Eingabegeräte wie Joystick (Gamecontroller im Allgemeinen) und Grafiktablett für den Entertainment- und Profidesignermarkt durchgesetzt. Die etablierten Interaktionsmethoden weisen aber immer noch Defizite auf. Da sie noch nicht intuitiv genug sind, müssen sie teilweise langwierig erlernt werden, was vielen Menschen immer noch schwer fällt. Die Entwicklung lässt jedoch erkennen, dass die Mensch-Maschine-Kommunikation zunehmend ergonomischer wird.

Genau hier setzt unter anderem die Forschung auf dem Gebiet der VR an. Man lässt zweidimensionale Repräsentationen am Bildschirm zunehmend hinter sich und versucht dreidimensionale Darstellungen zu schaffen, die eine weniger große Abstraktion darstellen und somit schneller und besser begreifbar sind. Viele der oben genannten Ein- beziehungsweise Ausgabemethoden und deren Geräte eigenen sich jedoch nur bedingt oder gar nicht für den Einsatz mit VR. Dies ist unter anderem auf die Klasse von Anwendungen zurückzuführen, für die sie konzipiert wurden. Um mit einem VR-System optimal arbeiten zu können, sollte nach BORMANN zum Beispiel „auch das Eingabegerät eine dreidimensionale Auflösung anbieten“ [Bor 94, Seite 49]. Da die virtuelle Realität auch, wie in Abschnitt 2.1 erwähnt, als die sechste Generation von Computern betrachtet werden kann, gibt es natürlich speziell entwickelte Aus- und Eingabegeräte für eine Fülle von Anwendungsgebieten. Die folgenden Unterkapitel beschäftigen sich daher eigens mit dieser Thematik.

2.3.1 Ausgabegeräte

2.3.1.1 Head Mounted Display

Head Mounted Displays (HMD) sind vom Prinzip her mit Brillen vergleichbar. Grund-

sätzlich unterscheidet man zwischen zwei verschiedenen Ausprägungen.

Bildschirmbrillen (Display-HMD) trennen den Benutzer vollständig von der reellen Umgebung und ermöglichen den Aufbau einer völlig neuen virtuellen Welt. Dazu wird vor jedes Auge ein separates LC-Display platziert. Je nach verwendeter Optik wird eine relativ große Leinwand in einiger Entfernung suggeriert. Wird die virtuelle Umgebung zudem in stereoskopischen Halbbildern präsentiert, kann dem Benutzer ein echter dreidimensionaler Eindruck vermittelt werden. Bildschirmbrillen bestechen damit durch einen recht hohen Grad an Immersion.

Durchsicht HMDs (See-Through-HMDs) haben die primäre Funktion die reelle Umgebung mit zusätzlichen Informationen aufzuwerten. Dafür sind vor den Augen halbdurchlässige Spiegel derart angeordnet, dass sie einerseits einen Blick auf die reale Umgebung und andererseits eine Überlagerung mit korrespondierenden computergenerierten Informationen ermöglichen. Neben der optischen Lösung mithilfe halbdurchlässiger Spiegel, existiert auch die Möglichkeit, die reale Umgebung mit am HMD befestigten Videokameras aufzunehmen. Die Videobilder werden dann mit virtuellen Objekten kombiniert.

Für immersive Anwendungen ist eine zur Szene passende Geräuschkulisse wichtig. Aus diesem Grund sind Head Mounted Displays oft mit Stereo-Kopfhörern ausgestattet. In heutige HMDs sind zudem Trackingsysteme integriert, welche es ermöglichen, Kopfposition und Blickrichtung des Benutzers zu ermitteln. Bei beiden HMD Systemen werden diese Informationen benutzt, um die virtuelle Darstellung entsprechend anzupassen. Daraus ergibt sich ein großer Vorteil von HMDs, der darin besteht, dass dem Benutzer ungeachtet von Kopfposition und Orientierung stets ein virtuelles Bild dargeboten werden kann. Ein gravierender Nachteil von HMDs besteht in der auftretenden Latenz zwischen schnellen Kopfdrehungen und korrespondierender Darstellung, die bis hin zu ernsthaften Wahrnehmungsstörungen führen kann (Simulator Sickness). Ebenfalls negativ zu bewerten sind die schlechte Bildqualität auf Grund geringer Auflösungen, das eingeschränkte Sichtfeld von durchschnittlich 40° sowie das Gewicht der Geräte, das bei längerem Tragen die Nackenmuskulatur belastet.

HMDs können vielfältig eingesetzt werden. In der Argumented Reality werden Durchsicht HMDs beispielsweise verwendet, um komplexe Arbeitsabläufe zu optimieren. Die einzelnen Arbeitsschritte können durch entsprechende Anleitungshinweise begleitet werden. Ein Beispiel dafür ist die Montage in der Automobilindustrie, wo dem Mitarbeiter komplexe Schaltpläne begleitend erläutert werden. Die stark immersiven Display HMDs eignen sich hervorragend für Trainings- oder Simulationsanwendungen, so zum Beispiel im militärischen Bereich für die Ausbildung von Kampfpiloten.



Abbildung 2-11: Display-HMD



Abbildung 2-12: Durchsicht-HMD

2.3.1.2 CAVE

Die erste Cave Automatic Virtual Environment (CAVE) wurde am Electronic Visualization Lab der University of Illinois in Chicago [CAVE] Anfang der neunziger Jahre entwickelt. Motivation war es, die Nachteile von HMDs zu überwinden.

Der Aufbau einer CAVE besteht aus drei bis fünf zu einem Raum angeordneten Rückprojektionswänden. Zur Minimierung des Platzbedarfes, kommen dabei Spiegel zum Einsatz. Um zudem den Boden auszuleuchten, kommt meist eine Aufprojektion hinzu. Der Raum ist so groß, dass sich sogar mehrere Benutzer gleichzeitg darin aufhalten können. Durch die hohe Abdeckung des Gesichtsfeldes ist die CAVE eines der immersivsten VR-Systeme. Unterstützt wird dies zusätzlich durch einen stereo-

skopischen Effekt, der bevorzugt mit Zeitmultiplexing oder Polarisation erzeugt wird. Benutzt man Zeitmultiplexing, müssen die stereoskopischen Halbbilder der einzelnen Wände synchron generiert und dargestellt werden. Früher wurden hierzu extrem teure Großrechner mit mehreren Grafikkarten eingesetzt, an welche CRT-Projektoren angeschlossen waren. Heute kommen zunehmend synchronisierte Cluster von Einzelrechnern und DLP-Projektoren zum Einsatz. Die zunehmende Verwendung von Standard-PC-Technik hat in den letzten Jahren zur Entwicklung verschiedener preisgünstiger CAVE-Systeme geführt. Das Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik (FIRST) war mit der X-Rooms-Technologie einer der Vorreiter, die diesen Ansatz verfolgten.

Die Darstellung innerhalb einer CAVE kann immer nur für eine diskrete Anzahl von aktiven Betrachterpositionen optimal erfolgen. Alle abweichenden Positionen führen durch die geometrischen Gegebenheiten des Raumes zu den für eine CAVE charakteristischen Verzerrungen in den Ecken. Durch die Bestimmung der Kopfposition eines Betrachters unter Verwendung eines Trackers, lässt sich die Darstellung der Szene korrekt und in Echtzeit anpassen. Für diesen Betrachter ist es dann möglich, sich innerhalb der CAVE zu bewegen, ohne das Verzerrungen auftreten. Zur Trennung der Ansichten mehrerer Betrachter (heute: maximal vier) bedarf es eines geeigneten Multiplexverfahrens. Zur Navigation innerhalb von virtuellen Welten kommen neben Trackingsystem vor allem Sprachsteuerungen zum Einsatz.

Die ursprüngliche CAVE wurde anfangs für die Simulation von Sternensystemen und Wetterphänomenen eingesetzt. Heutige Anwendungsgebiete der CAVE umfassen die Entwicklungsunterstützung in der Automobilindustrie oder Gebäudeplanung (Architektur). An vielen amerikanischen Universitäten werden CAVEs zur anschaulichen Darstellung dreidimensionaler Funktionen der Mathematik und für weitere Bildungsbereiche eingesetzt.



Abbildung 2-13: CAVE Installation bei FIRST

2.3.1.3 Workbench

Die Idee und der erste Prototyp einer Workbench entstanden 1994 bei der GMD[2]. Die an einen Tisch erinnernde *Responsive Workbench* besteht im Wesentlichen aus einer horizontal angeordneten Rückprojektionsscheibe, welche von unten über einen Spiegel ausgeleuchtet wird. Für einen räumlichen Eindruck werden stereoskopische Halbbilder mittels Zeitmultiplexing voneinander getrennt. Analog der CAVE wird es dadurch möglich, mehrere Benutzer einzubeziehen. Um eine betrachterbezogene Darstellung zu errechnen, erfassen auch bei der Workbench Tracking-Sensoren die Kopfpositionen und -orientierungen der Nutzer.

Im Gegensatz zu HMD und CAVE besitzt die Workbench weniger Positionen aus denen eine Sicht auf die virtuelle Welt möglich ist. Ihre Vorteile liegen in der Interaktion mit Objekten, da jeder Teil in unmittelbarer physischer Reichweite liegt und mit der Hand greifbar scheint. Aus diesem Grund kommen bei Worbenches gehäuft Daten-

[2]Die GMD – Forschungszentrum Informationstechnik GmbH wurde erst vor kurzem in die Fraunhofer Gesellschaft eingegliedert.

handschuhe und Stifte zum Einsatz (siehe Abschnitt 2.3.3.1, 3.5). Heutige Workbenches sind recht flexibel. Hersteller wie Fakespace Systems bieten Geräte an, die sich teilweise bis in die Vertikale kippen lassen oder sogar aus zwei zu einem „L“ geformten Projektionsflächen bestehen.

Für die Workbench gibt es einige interessante Anwendungsmöglichkeiten. So wird sie zum Beispiel für das schnelle Entwickeln von architektonischen Prototypen oder zur Flächenplanung eingesetzt.



Abbildung 2-14: Barco Baron Workbench

2.3.1.4 PowerWall

Powerwalls sind überaus große und hochauflösende Projektionswände, die oft auch eine leicht gekrümmte Form aufweisen. Aufgrund dessen zeichnen sie sich durch einen hohen Immersionsgrad aus. Ihre Größe und hohe Auflösung erreichen sie durch den Einsatz von mehreren neben- und untereinander angeordneten Projektionen, die sich an den Rändern leicht überlappen. Mithilfe spezieller Blending-Software werden

die Projektionen aufeinander abgestimmt und ergeben ein optimales Gesamtbild.



Abbildung 2-15: Barco BR Center

2.3.1.5 Volumetrische Displays

Alle bisher betrachteten stereoskopischen Systeme erschaffen die Illusion von Dreidimensionalität durch Projektion auf zweidimensionale Flächen. Einen ganz anderen Ansatz verfolgen volumetrische Displays. Sie besitzen die Fähigkeit Punkte im Raum direkt zu illuminieren und gehören damit zu den autosteroskopischen Displays. Daraus folgt, dass es der räumlichen Betrachtung dargestellter Objekte keiner Brille bedarf. Des Weiteren ist die Anschauung der Objekte aus unterschiedlichen Perspektiven ohne jegliche Anpassung der Darstellung möglich. Das heißt, ein beziehungsweise mehrere Betrachter können um das Objekt herumgehen. Eine kollaborative Arbeit an Objekten wird somit erleichtert.

Das Funktionsprinzip dieser Displays basiert auf schnell rotierenden Flächen, die ein durch eine Sichtkuppel geschütztes Volumen überstreichen. Zu einem Zeitpunkt nimmt die Fläche dabei eine durch ihre Form bestimmte Punktmenge des Volumens ein. Durch Anstrahlen der Fläche werden die Punkte für den Moment sichtbar. Das Felix 3D-Display [Bahr 96] benutzt eine helixförmige Fläche. Über eine spezielle Optik

können Laserstrahlen umgelenkt werden und erhellen einen Voxel, wenn der Strahl im entsprechenden Moment auf die Helix trifft. In einer schnellen zeitlichen Abfolge entsteht dann auf Grundlage eines 3D-Modells dessen echte räumliche Repräsentation. Ganz ähnlich funktioniert der Perspecta von Actuality Systems [Fav 02], nur nutzt das Gerät einen planaren rotierenden Schirm und normale Projektionstechnologie anstatt eines Lasers. Aus seiner dreidimensionale Auflösung von 768x768x198 ergibt sich eine Anzahl von über 100 Millionen Voxeln. Wollte man das 3D-Bild in heute üblichen 24 Bit Farbtiefe darstellen, bedeutete dies einen immens hohen Bedarf an schnellem Bildspeicher. Aus technischen Gründen geht man beim Perspekta daher einen Kompromiss zu Lasten der Farbtiefe ein, der Farbraum ist nur drei Bit breit. Diesen Umstand versucht man aber mithilfe von Dithering-Verfahren auszugleichen.

Ein weiterer die Immersion trübender Nachteil volumetrischer Displays ist deren teils technisch bedingte Größe. Mit maximal 50 Zentimetern Durchmesser sind sie im Vergleich zu anderen VR-Systemen sehr klein.



Abbildung 2-16: Perspecta von Actuality Systems

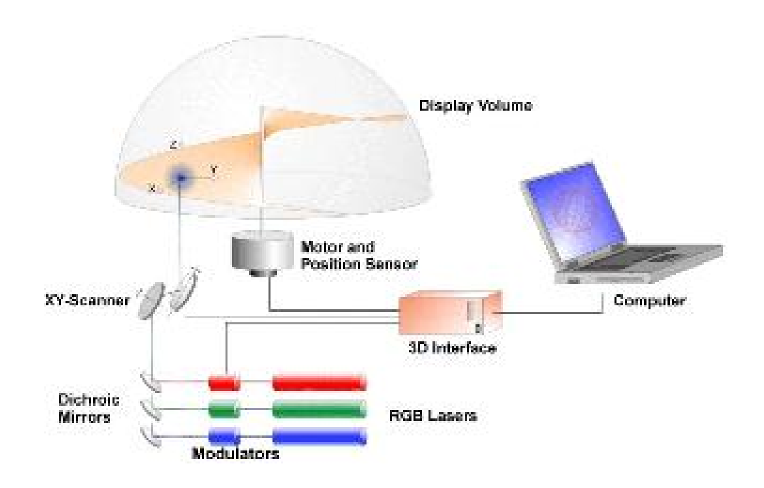


Abbildung 2-17: Felix 3D Funktionsprinzip

2.3.2 Tracking

Tracking[3] beschreibt Methoden der Akquisition von dreidimensionalen Positions- und Orientierungsdaten, die als Eingabe für VR-Systeme eine wichtige Rolle spielen. Möchte man mit einer virtuellen Umgebung interagieren, sind Informationen über die Position und Lage von Kopf, Händen und des restlichen Körpers einer Person von entscheidender Bedeutung. Dieser Abschnitt gibt eine Übersicht über die wichtigsten angewendeten Trackingverfahren. Er beschreibt wie sie funktionieren und zeigt dabei Vor- und Nachteile der einzelnen Verfahren auf.

Tabelle 2-3 gibt eine Übersicht wichtiger Kenngrößen, die alle Positionssensoren gemein haben und anhand derer sie sich vergleichen lassen.

[3] engl. *to track* = verfolgen

Eigenschaft	Beschreibung
Auflösung	kleinster wahrgenommener Positionswechsel
Genauigkeit	räumliche Abweichung zwischen gemessener und tatsächlicher Position
Aktualisierungsrate	Zeit zwischen den einzelnen Positionsmeldungen
Verzögerung	Zeit zwischen Positionswechsel und der Meldung der neuen Position
Wiederholbarkeit	Abweichung zwischen Start- und Endpunkt nach Vollführung einer Schleifenbewegung

Tabelle 2-3: Eigenschaften von Tracking-Systemen [Bor 94, Seite 57]

Der Verzögerung ist dabei ein besonderer Stellenwert beizumessen. Eine Latenz von mehr als 1/6 Sekunden zwischen physischer Bewegung und korrespondierender visueller Rückkopplung zerstört nicht nur die Illusion der Immersion [Bor 94, Seite 94-95], sondern kann auch das Wohlbefinden des Benutzers beeinträchtigen. Dieses Phänomen, das bis hin zu Übelkeit oder Erbrechen führen kann, wird auch als Simulatorschwindel[4] bezeichnet. Simulatorschwindel und die Schiffs- beziehungsweise Reisekrankheit haben den gleichen Ursprung. Ursache ist ein Konflikt im Gehirn, das versucht, zwei sich unterscheidende Sinneseindrücke in Einklang zu bringen. Die visuelle Wahrnehmung weicht hierbei maßgeblich von der Information ab, die der Vestibularapparat (Gleichgewichtssinn) dem Gehirn meldet. Simulatorschwindel ist ein ernstzunehmendes Problem, dessen Kompensation sich viele Arbeiten in diesem Bereich widmen.

Man unterscheidet zwischen aktivem und passivem Tracking. Aktive Systeme bestehen aus einem Sender und Sensoren. Der Sender emittiert ein Signal, das die Sensoren aufnehmen. Aus der Änderung zum Ursprungssignal kann dann mithilfe kom-

[4] engl. *Simulator Sickness*

plizierter mathematischer Algorithmen die Position des Sensors relativ zum Sender berechnet werden. Bei passiven Trackingverfahren sind hingegen nur Sensoren erforderlich. Im Gegensatz zum aktiven Tracking benötigt man kein künstlich erzeugtes Signal, sondern bedient sich schon existierender Gegebenheiten. Als Beispiel hierfür können Beschleunigungsmesser angeführt werden. Da sie keinen festen Bezugspunkt besitzen, funktionieren sie additiv. Dadurch ergibt sich jedoch der Nachteil einer sich akkumulierenden Fehlerfortpflanzung.

Eine weitere Charakteristik von Tracking-Systemen stellt die Anzahl von zur Verfügung gestellten Freiheitsgraden dar. Während sogenannte 3-DOF[5]-Systeme entweder nur Orientierungen oder nur Positionen liefern können, sind 6-DOF-Systeme in der Lage beides gleichzeitig zu leisten.

Neben den im Folgenden vorgestellten Verfahren zur räumlichen Positions- und Lagebestimmung existieren noch andere Formen des Tracking. Der Vollständigkeit halber soll hier das Eye-Tracking Erwähnung finden, welches anhand bildverarbeitender Ansätze versucht, die Blickrichtung von Benutzern zu bestimmen und für VR-Anwendungen nutzbar zu machen.

2.3.2.1 Mechanische Tracking-Systeme

Der älteste Ansatz zur Positionsbestimmung benutzt mehrere durch Gelenke aneinander gekoppelte Arme. Das erste System dieser Art testete Ivan Sutherland[6] bereits in den sechziger Jahren. Der gesamte Aufbau eines mechanischen Tracking-Systems ist meist an der Wand oder an der Decke verankert. Der Benutzer steckt seinen Kopf oder seine Hand in einen ergonomischen Aufsatz am Endstück der Konstruktion. Um die Position zu ermitteln, misst man zunächst die Winkelstellungen der Gelenke. Zusammen mit den Längen der einzelnen Verbundstücke ergeben sich Transformationsmatrizen, mit deren Hilfe sich die relative Position des Benutzers zur Basis des Systems

[5] engl. *degree of freedom* (DOF) = Freiheitsgrad

[6] Ivan Sutherland studierte am Massachusetts Institute of Technology (MIT) und gilt als Pionier auf dem Gebiet der interaktiven Computergrafik

bestimmen lässt.

Dieser Ansatz des Trackings zeichnet sich durch seine hohe Genauigkeit und eine besonders kleine Verzögerung aus. In den Spezifikationen entsprechender Geräte ist von einer Genauigkeit von vier Millimetern und 0,1° bei nur einer Millisekunde Verzögerung die Rede. Weitere Vorteile mechanischer Systeme liegen in der Kombinationsmöglichkeit mit Kraftrückkopplungsmechanismen (Force Feedback) und einer relativ hohen Reichweite bei entsprechend großen Ausführungen. Nachteilig zu sehen sind die hohen Kosten und der durch die Konstruktion bedingte doch etwas eingeschränkte Bewegungsspielraum beim Benutzer.



Abbildung 2-18: Phantom von SensAble (Interaktionstracking mit Force Feedback)



Abbildung 2-19: Boom von Fakespace Labs (Betrachter-Tracking)

2.3.2.2 Akustische Tracking-Systeme

Akustische Tracking-Systeme nutzen Ultraschallemitter und Mikrofone für eine Entfernungsbestimmung. Die Schallgeschwindigkeit ist eine feste und bekannte Größe. Durch einfache Multiplikation mit der gemessenen Laufzeit zwischen Aussenden und Empfangen des Schallsignals kann die Distanz zwischen Sender und Empfänger ermittelt werden. Prinzipiell ist es ohne Bedeutung, ob der Sender oder Empfänger dabei fest positioniert und das entsprechende Gegenstück im Raum bewegt wird. In vielen Fällen sind jedoch drei Ultraschallmikrofone, in einer bestimmten Anordnung (z.B. rechtwinkliges Dreieck), zu einer Basisstation kombiniert. Hieraus ergeben sich drei Laufzeitmessungen, aus denen mittels Triangulation die Position des Senders geschlussfolgert werden kann. Um zusätzlich die Orientierung im Raum feststellen zu können, sind ebenfalls drei zu einer Einheit verbundene Sender notwendig.

Akustische Systeme bestechen durch ihren geringen Kostenaufwand und einen recht großen Aktionsradius. Im Gegensatz dazu steht die Anfälligkeit gegenüber Hintergrundgeräuschen und von Flächen reflektierten Echos des ausgesendeten Schallimpulses. Die so entstehenden Interferenzen beeinträchtigen eine einwandfreie Messung und gehen daher mit Genauigkeitsverlusten einher. Des Weiteren pflanzen sich

die hohen Frequenzen von ca. 23KHz unidirektional aus. Daher ist eine direkte Sichtverbindung zwischen Sender und Empfänger unabdingbar. Ebenfalls macht sich der hohe Schalldruck nachteilig bemerkbar.



Abbildung 2-20: Logitech Fly Mouse

2.3.2.3 Optische Tracking-Systeme

Die Grundlage dieses Verfahrens bilden optische Markierungen (Marker) die von Kameras aufgezeichnet werden. Als Marker dienen speziell reflektierende Materialen, Karten mit besonderen Form- beziehungsweise Farbmustern oder Leuchtdioden. Eine jeweils spezielle Bildanalysesoftware ist in der Lage, in den Kameraaufnahmen die Marker zu identifizieren und daraus Rückschlüsse auf deren Position und Lage im Raum zu ziehen. Analog zu akustischen Systemen können Kameras entweder am beweglichen Objekt oder stationär befestigt werden. Im ersten Fall genügt oft eine Kamera, die auf die meist an der Decke angebrachten Marker gerichtet ist. Im zweiten Fall benötigt man mindestens zwei Kameras, aus deren Aufnahmen mithilfe von Triangulation die Position eines frei beweglichen Markers bestimmt wird.

Obwohl optische Systeme als überaus exakt gelten, besitzen sie einen entscheidenden Nachteil, unter dem auch schon akustische Systeme zu leiden haben. Werden Marker von Körperteilen oder anderen Gegenständen verdeckt, kann keine Positions-

bestimmung vorgenommen werden. Um das Verdeckungsproblem zu minimieren, benutzt man daher viele Kameras und Marker. Dies hat jedoch wiederum zur Folge, dass der Berechnungsaufwand unverhältnismäßig ansteigt. Um auftretende Messlücken zu überbrücken, werden akustische und vor allem optische Verfahren oft hybrid mit inertialen Tracking-Geräten wie Gyroskopen und Beschleunigungsmessern betrieben.

Neben den Marker basierten Systemen existieren auch markerlose optische Verfahren, welche in Bildern die Silhouette von Objekten oder Menschen erkennen können. Zu ihnen gehört beispielsweise das für die Sony Playstation 2 entwickelte Eye Toy. Die Methode eignet sich jedoch nicht, um räumliche Informationen zu gewinnen.



Abbildung 2-21: Infrarot Kamera der A.R.T. GmbH



Abbildung 2-22: Hand-Marker der A.R.T. GmbH

2.3.2.4 Elektromagnetische Tracking-Systeme

Elektromagnetisches Tracking basiert auf dem physikalischen Prinzip der Induktion. Dessen Quintessenz ist, dass ein stromdurchflossener Leiter stets auch von einem elektromagnetischen Feld umgeben wird. Gleichermaßen wird in einen Leiter, der sich in einem elektromagnetischen Feld befindet, ein Strom induziert.

Sender und Empfänger solcher aktiven Systeme bestehen jeweils aus drei senkrecht zueinander angeordneten Spulen. In einem Zyklus wird nacheinander Spannung an alle drei Spulen des Senders angelegt. An den drei Spulen der Sensoren wird dann

jeweils der induzierte Strom gemessen. Auf Grund der Tatsache, dass das Feld parallel zu einer stromdurchflossenen Spule stark und senkrecht zu ihr eher schwach ausfällt, differieren diese Ströme. Insgesamt neun Messungen pro Zyklus benötigt also ein komplizierter mathematischer Algorithmus für die Positions- und Orientierungsbestimmung eines Empfängers. Die Anzahl der Messzyklen pro Empfänger bewegen sich zwischen 30 und 120Hz.

Die Leistung auf dem Markt befindlicher Systeme unterscheidet sich teilweise beträchtlich. Beispielsweise variiert die Reichweite zwischen einem und neun Metern und die Anzahl einsetzbarer Sensoren zwischen einem und zwanzig. Die Genauigkeit der Geräte nimmt mit zunehmender Entfernung zum Sender ab. Elektromagnetische Trackingsysteme haben zudem einen weiteren großen Nachteil. Andere Strahlungsquellen sowie ferromagnetische Metallelemente in der Umgebung deformieren das elektromagnetische Feld und verfälschen somit die resultierenden Daten. Positiv zu bewerten ist jedoch die Tolleranz gegenüber nichtmetallischen Hindernissen im Raum zwischen Sender und Empfänger.



Abbildung 2-23: Nest of Birds von Acension Technology

2.3.3 Eingabegeräte

2.3.3.1 Datenhandschuhe

Die Hand stellt das wichtigste Interaktionswerkzeug des Menschen dar, mit ihr werden die unterschiedlichsten alltäglichen Aufgaben bewältigt. Es erscheint daher sinnvoll, natürliche Handbewegungen als Eingabe für VR-Systeme zu nutzen. Datenhandschuhe lassen sich meist wie normale Handschuhe über die Hand streifen. Sie sind in der Lage, die Stellung der Finger zu messen und in für einen Computer lesbare Signale umzuwandeln. Grundsätzlich unterscheidet man zwischen zwei Kategorien von Datenhandschuhen [Bow 04, Seite 106ff].

Bend-Sensing Gloves Zur ersten gehören Geräte, die die Fähigkeit besitzen den Beugungsgrad einzelner Finger zu bestimmen. Um dies zu bewerkstelligen, haben sich unterschiedliche Verfahren bewährt.

Das erste hüllt die Hand in ein mechanisches Exoskelett. Die Fingerglieder sind dabei durch Scharniere miteinander verbunden, deren Stellung mittels spezieller Sensoren bestimmt werden kann. Daraus lässt sich der Winkel zwischen zwei Fingergliedern überaus exakt ableiten.

Eine kostengünstige Methode wurde Ende der achtziger Jahre für den PowerGlove entwickelt. In den Fingern des Handschuhs sind Streifen untergebracht, die mit einer besonderen Flüssigkeit imprägniert sind. Biegt sich ein Streifen infolge einer Fingerbewegung, ändert sich der elektrische Widerstand der Flüssigkeit. Dieses Verfahren gilt als verhälnismäßig unpräzise.

Ein weiteres alternatives Verfahren basiert auf Licht. Hierzu sind für jeden Finger Glasfaserschlaufen in den Datenhandschuh eingearbeitet. An einem Ende emittiert eine Leuchtdiode Licht in die Glasfaser, am anderen Ende misst ein Photodetektor die Menge des ankommenden Lichts. Auf dem Weg durch den Lichtleiter geht in Abhängigkeit vom Beugungsgrad des Fingers Licht verloren. Es tritt also eine messbare Diskrepanz zwischen ausgesendetem und empfangenen Licht auf, die auch als

optischer Widerstand bezeichnet werden kann.

Pinch Gloves Die zweite Kategorie bilden Datenhandschuhe, die lediglich ermitteln können, ob sich zwei oder mehrere Fingerspitzen berühren. Dazu sind leitfähige Kontaktstellen beziehungsweise -streifen in die Oberfläche, speziell an den Fingerspitzen, des Handschuhs eingearbeitet. Berühren sich die Kontaktstellen wird ein Stromkreis geschlossen, der einer entsprechenden Fingerkombination zugeordnet werden kann.

Pinch wie Bend-Sensing Gloves ermöglichen generell aus der Kombination von Fingerstellungen, bestimmte Gesten abzuleiten, die in VR-Applikationen verschiedenen Kommandos zugeordnet werden können. Bend-Sensing Gloves erlauben durch ihre stufenlose Auflösung aber auch filigrane Greifoperationen. Um eine Gesten- oder Greifoperation im Kontext von Lage und Position der Hand richtig deuten zu können, werden Datenhandschuhe in der Regel mit Tracking-Systemen kombiniert. Die Firma Immersion hat mit dem CyberForce einen Datenhandschuh entwickelt, dessen Position mittels mechanischem Tracking erfasst wird. Gleichzeitig ist der komplette Aufbau mit unterschiedlichen (ground-, body-referenced) Kraftrückkopplungsmechanismen (Force Feedback) ausgestattet, sodass das Greifen virtueller Objekte auf natürliche Art und Weise simuliert werden kann.



Abbildung 2-24: 5DT Bend-Sensing Glove



Abbildung 2-25: Fakespace Labs Pinch Gloves

2.3.3.2 Wands/3D-Mäuse

Wands[7] beziehungsweise 3D-Mäuse bilden eine eigene Klasse von Eingabegeräten, die in der Hand gehalten oder an ihr fest gemacht werden können. Es gibt sie in kabellosen wie kabelgebundenen Ausführungen. Sie besitzen oft eine Vielzahl programmierbarer Knöpfe und sind mit Tracking-Systemen kombiniert. Dabei variiert die Anzahl der Freiheitsgrade zwischen drei und sechs. Zu den ersten Wands gehörte auch der Poolball von Polhemus, der schon in der ersten CAVE Anwendung fand.

Space- sowie Gyromäuse sind zwar keine echten 3D-Mäuse, sollen hier aber trotzdem Erwähnung finden. Die Space Mouse besitzt eine Reihe von Knöpfen und einen Dial mit sechs Freiheitsgraden. Sie wird bevorzugt bei der Modellierung in CAD-Anwendungen genutzt. Die Gyromaus besitzt ebenso viele Freiheitsgerade wie eine klassische Maus. Im Gegensatz zur klassischen Maus generiert sie diese aber nicht durch die Bewegung auf einer planaren Oberfläche, sondern durch Schwenkbewegungen im Raum, die von einem Gyroskop erfasst werden. Gyromäuse sind heute meist kabellos und werden ähnlich einer Fernbedienung in der Hand gehalten.



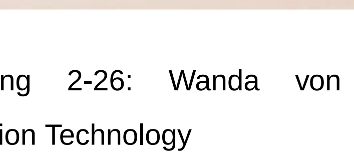


Abbildung 2-26: Wanda von Ascension Technology



Abbildung 2-27: Flystick der A.R.T. GmbH

[7] engl. *wand* = Zauberstab

2.3.3.3 Gamecontroller

Gamecontroller wie Joysticks oder Gamepads besitzen eine lange Tradition im Einsatz für Computerspiele. Die klassischen Beispiele sind Renn- und Flugsimulationen. Sie besitzen viele Knöpfe, sind teils kabellos und werden heute zunehmend mit Force Feedback ausgestattet. Während Joysticks sogar gerichtete Kräfte erzeugen können, geben Gamepads nur eine Rückkopplung mittels Vibration. Joysticks und Gamepads werden hauptsächlich zur Fortbewegung in virtuellen Szenarien genutzt.



Abbildung 2-28: Logitech Freedom 2.4 Cordless Joystick

2.3.4 Software

Im Laufe der Zeit wurde eine Vielzahl von VR-Software-Systemen entwickelt. Die Programmierbibliotheken und -schnittstellen sind so konzipiert, dass sich der Programmierer auf die Entwicklung der VR-Anwendungen konzentrieren kann, ohne mit Details eines Betriebssystems oder Schwierigkeiten der VR selbst in Berührung zu kommen. Die Bibliotheken ermöglichen von vornherein Dinge wie Viewporterstellung, betrachterbezogene perspektivische Projektion, stereoskopische Darstellung oder die Synchronisation eines Clusters. Des Weiteren integrieren sie bereits die Ansteuerung verschiedenster Tracking- und anderer Eingabegeräte. Auch eine Ausgabe auf verschiedenen Systemen wie CAVE, HMD oder Workbench ist mit ihnen möglich. Zur Visuali-

sierung nutzen die Systeme echtzeitfähige 3D-Grafik-Schnittstellen wie OpenGL oder Direct3D. Die Manipulation der virtuellen Welt erfolgt meist auf der Grundlage eines Szenengraphen. Es gibt sowohl kommerzielle wie auch frei verfügbare Implementierungen. Zu den bekanntesten Vertretern gehören die CAVELib und das WorldToolkit. Das VR-Software-Projekt DIVERSE (Device Independent Virtual Environments- Reconfigurable, Scalable, Extensible) hält unter folgender URL eine ausführliche Liste von VR-Software-Systemen zum Abruf bereit:

http://www.diverse.vt.edu/VRSoftwareList.html (letzter Abruf: 29.10.2004)

2.3.5 Anwendungsgebiete

Der heutige technische Stand erlaubt der VR, den Benutzer in beliebige Szenarien hinein zu versetzen. Im Laufe der Entwicklung hat sich die VR aber für einige spezielle Anwendungsgebiete als besonders wertvoll herausgestellt. Zu diesen gehören die Bereiche Simulation und Training, Ausbildung, Unterhaltung, Kunst, Visualisierung, Architektur, Medizin und interaktive Zusammenarbeit. Im folgenden sollen einige spezielle Anwendungen näher gebracht werden.

In der Automobilindustrie wird VR in den letzten Jahren verstärkt eingesetzt, um vor allem die Kosten während der Entwicklung von neuen Modellen zu senken. Bevorzugt an Powerwalls werden Design Reviews vorgenommen, um einen ersten echten Eindruck des am Computer entworfenen Autos zu erhalten. Eventuelle Mängel können dadurch leichter und im Entwicklungsprozess eher erkannt und beseitigt werden. Auf der Grundlage physikalischer Gesetzmäßigkeiten wird zum einen das Fahrverhalten des Autos in Fahrsimulatoren getestet. Zum anderen können Crashtest durchgeführt werden, noch bevor der erste Prototyp gebaut worden ist.

Vor allem das amerikanische Militär setzt VR zum Manövertraining ein. Hierzu werden eine große Anzahl verschiedener Simulatoren für Schiffe, Flugzeuge und Panzer miteinander vernetzt. Mithilfe von HMDs und Laufgeräten kann auch die Infanterie in die simulierte Übung einbezogen werden.

Im Verkehrswesen wie Eisenbahn, Luft- und Schifffahrt werden zu Ausbildungszwecken Übungen an entsprechenden Simulatoren durchgeführt. Des Weiteren werden Nofallsituationen durchgespielt, um im Ernstfall schnell reagieren zu können.

In der Architektur findet die VR breite Anwendung. Walkthrougs gestatten eine anschauliche Präsentation, da sie eine virtuelle Begehung bestehender oder erst im Entwurf existierender Gebäude ermöglichen. Des Weitern können mithilfe von Simulationen die Lichtverhältnisse zu verschiedenen Tageszeiten visualisiert werden. Im Facility Management wird mittels VR das Fluchtverhalten bei Notfällen untersucht, um Fluchtpläne zu optimieren.

2.3.6 Interaktion

Die Interaktion mit virtuellen Welten unterscheidet grundsätzlich drei Aspekte. Diese sind Fortbewegung, Selektion beziehungsweise Manipulation und Systemsteuerung. Im Folgenden sollen die in VR-Systemen häufig verwendeten Interaktionsparadigmen und Metaphern der drei Interaktionsaspekte vorgestellt werden.

2.3.6.1 Fortbewegung

Die Fortbewegung innerhalb einer virtuellen Welt beschreibt grundsätzlich den Vorgang von der aktuellen Position zu einem anderen gewünschten Ort zu gelangen und bildet damit die elementarste aller Interaktionen. Da eine Fortbewegung meist sekundären Charakter besitzt und zur Lösung einer höheren Aufgabe dient, müssen die Methoden einfach und so intuitiv wie möglich sein. Zudem spielen für die Bewegung in virtuellen Welten verschiedene Aspekte eine Rolle. So zum Beispiel die zu überbrückende Distanz, die Anzahl von Neuorientierungen auf dem Weg zum Ziel, ob das Ziel vom Startpunkt sichtbar ist oder die benötigte Anzahl von Freiheitsgraden beziehungsweise Genauigkeit für eine Bewegung. Des Weiteren kann eine Fortbewegung passiven oder aktiven Charakter besitzen.

Passive Bewegungen werden vom System kontolliert, so zum Beispiel animierte

Kamerafahrten zwischen definierten Sichtpunkten.

Von weitaus größerem Interesse sind jedoch aktive Bewegungen, bei denen der Benutzer die Steuerung des virtuellen Sichtpunktes bestimmt. Die Motivation einer vom Benutzer gesteuerten Bewegung kann dabei eine Erkundung der Umgebung oder das Aufsuchen eines bestimmten Zieles sein. Um Bewegungen aktiv zu steuern, müssen unterschiedliche physische (echte) Bewegungen mithilfe verschiedener Methoden und Eingabegeräte auf eine virtuelle Fortbewegung abgebildet werden. Diese Abbildung kann auf eine mehr oder weniger natürliche Art und Weise geschehen. Man unterscheidet daher zwischen Methoden, die eine *natürliche Entsprechung* besitzen, und *abstrakten Metaphern*.

Das Betrachter-Tracking ist die einzige Technik, bei der eine physische Translation oder Rotation des Körpers des Benutzers unmittelbar in eine virtuelle Bewegung umgesetzt wird. Alle anderen Techniken sind immer mit einer gewissen Abstraktion verbunden.

Bei simulierten physischen Bewegungen kommen spezielle Geräte (Locomotion Devices) wie Laufbänder, Stepper oder fahrradähnliche Hometrainer zum Einsatz. Physisch gesehen bleibt der Benutzer an Ort und Stelle. Eine auf das Gerät übertragene Bewegung kann aber in eine entsprechende Fortbewegung in der virtuellen Welt übersetzt werden.

Ein weitere Methode wird als „Grabbing the Air“ bezeichnet. Hierzu werden Datenhandschuhe eingesetzt. Der Benutzer greift an eine Position und bewegt daraufhin seine Hand. Das Resultat ist eine entsprechende Bewegung durch die virtuelle Welt. Die Technik ist vergleichbar mit dem Entlanghangeln an einem imaginären Seil.

Simulierte Fahrzeuge wie Autos oder Flugzeuge eignen sich gut zur Überwindung großer virtueller Distanzen und haben den Vorteil, dass sie natürliche Entsprechungen besitzen. Der Benutzer bleibt wieder stationär und sitzt in einem virtuellen oder in einer nachempfundenen Kulisse des Führerraums des jeweiligen Fahrzeuges. Mit Hydraulik ausgestattete Fahr- und Flugsimulatoren, können sogar auftretende Beschleunigungs-

kräfte erzeugen. Zur Steuerung der Fahrzeuge werden teils mit einer Kräfterückkopplung versehene Eingabegeräte wie Joysticks oder Lenkräder genutzt.

Zur Klasse von Fortbewegungsmöglichkeiten ohne natürliche Entsprechung gehören abstrakte Metaphern wie die World-in-Miniature Technik [Pau 95]. Dabei wird über der normalen Ansicht auf die virtuelle Welt eine skalierte Gesamtansicht eingeblendet, die eine exakte Kopie dieser darstellt. Innerhalb der Miniaturansicht wird ein die aktuelle Kameraposition repräsentierender Avatar angezeigt. Dieser lässt sich an unterschiedliche Positionen der Miniaturwelt verschieben und ist ein Beispiel dafür, dass Selektion beziehungsweise Manipulation virtueller Objekte (siehe unten), in diesem Fall des Avatars, und Fortbewegung einander bedingen können. Die Folge ist eine Animation, die die aktuelle Kameraposition an die des Avatars verschiebt. Dies kann durch eine Kamerafahrt in der großen Ansicht geschehen. Eine andere Variante ist der Flug in die Miniaturwelt, sodass diese zur großen Ansicht wird. Nach Ablauf der Animation wird eine neue Miniaturwelt eingeblendet.

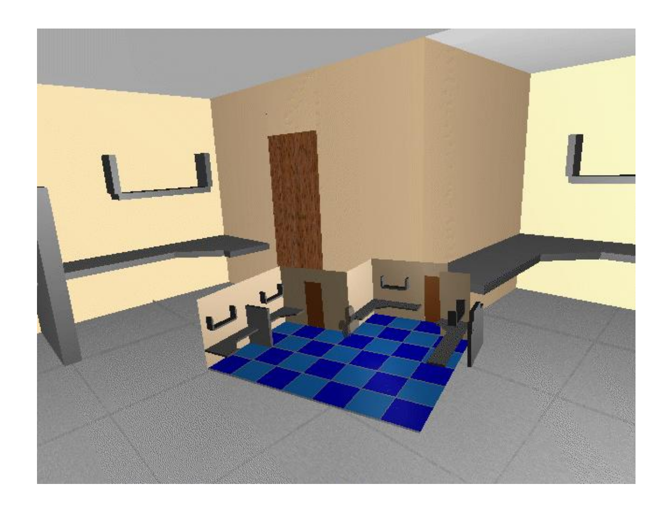


Abbildung 2-29: World-in-Miniature

Lenktechniken sind intuitiv, besitzen aber ebenfalls keine natürliche Entsprechung. Sie beschreiben eine Bewegung entlang von Vektoren, die vom menschlichen Körper

definiert werden. So zum Beispiel Blickrichtung, Oberkörperausrichtung oder die Vektoren entlang der Arme. Sie lassen sich mithilfe von Tracker erfassen und in das Koordinatensystem der virtuellen Welt transformieren. Durch Auslösen eines diskreten Ereignisses (Drücken eines Knopfes an einem entsprechenden Eingabegerät oder durch Datenhandschuh erkannte Geste) wird eine Bewegung mit einer definierten Geschwindigkeit entlang des entsprechenden Vektors gestartet. Ein anderes Ereignis (Knopf loslassen, Auflösen der Handgeste) stoppt die Bewegung.

2.3.6.2 Selektion und Manipulation virtueller Objekte

Eine Selektion beziehungsweise Manipulation von Objekten innerhalb einer virtuellen Welt kann *direkt* oder *indirekt* erfolgen. Man spricht auch von gleichförmigen (isomorphen) beziehungsweise ungleichförmigen (nicht-isomorphen) Techniken. Eine direkte Methode setzt voraus, dass sich ein virtuelles Objekt in unmittelbarer physischer Reichweite befindet. Durchgeführte Operationen werden also eins-zu-eins abgebildet und entsprechen damit Manipulationen von Objekten in der natürlichen Welt. Der Handlungsspielraum ist dabei aber oft begrenzt. Technische Parameter wie zum Beispiel die Reichweite von Tracking-Systemen, rein physische Grenzen (Wände einer CAVE) oder ganz einfach die Armlänge des Benutzers scheinen unüberwindbare Barrieren zu sein. Infolge dessen wurden nicht-isomorphe Methoden entwickelt, die Selektion und Manipulation zwar auf eine abstrakte Ebene verlagern, isomorphen Methoden bezüglich Nutzbarkeit und Performanz aber in nichts nachstehen.

Eine weitere Klassifikation kann nach ego- beziehungsweise exozentrischen Gesichtspunkten, das heißt, ob man die Welt aus eigener oder höherer Perspektive sieht, erfolgen. Zu den exozentrischen Methoden gehört die bereits erwähnte World-in-Miniature Metapher, die nicht nur zur Fortbewegung in virtuellen Welten angewendet wird. Neben dem Avatar lassen sich nämlich auch Objekte der Miniaturwelt verschieben. Alle Objekte können dabei isomorph manipuliert werden.

In der natürlichen Welt erfolgt eine Selektion von Objekten durch Anfassen. Entsprechend spürt man das Objekt in der Hand. In der virtuellen Welt kann die Selektion wie oben beschrieben direkt oder indirekt erfolgen.

Direkt gesehen geschieht dies, indem die Hand zur Position des virtuellen Objektes geführt wird. Bei komplett abschirmenden VR-Systemen wie dem Display-HMD, in denen der Benutzer seine Hand selbst nicht sehen kann, ist hierfür eine virtuelle Repräsentation der Hand notwendig, die alle Bewegungen der echten Hand nachempfindet. Diese Technik wird auch als Virtual Hand bezeichnet.

Indirekt geschieht die Selektion eines Objektes, indem man darauf zeigt. Dies erfordert einen Zeigevektor, der vorrangig aus Tracking-Informationen gewonnen wird. Er kann beispielsweise die Verlängerung eines Armes bilden. Des Weiteren kann der Vektor aus Handposition und -orientierung, analog auch aus Kopfposition und -orientierung (Blickrichtung), Kopf- und Handposition sowie den Positionen und Orientierungen beider Hände abgeleitet werden.

Ein Zeigevektor kann zudem unterschiedliche Merkmale aufweisen. So definiert seine Länge beispielsweise einen diskreten Aktionsradius. Der Vektor kann entweder einen Selektionsstrahl oder ein Selektionsvolumen beschreiben. Ein Selektionsvolumen ähnlich einem Taschenlampenstrahl, der sich kegelförmig ausbreitet, besitzt den Vorteil, dass kleine und weit entfernte Objekte leichter erfassbar sind. Der Nachteil besteht darin, dass sich eine eindeutige Selektion bei dicht beieinander liegenden Objekten eher schwierig gestaltet. Des Weiteren kann der Vektor über Flexibilität verfügen und einen gekrümmten Verlauf anzeigen, wodurch auch teilweise verdeckte Objekte selektiert werden können.

Die verschiedenen Varianten des Vektors erfordern unterschiedliche Repräsentationen. Er kann als komplett durchgezogener Strahl oder als 3D-Cursor am Endpunkt visualisiert werden.

Mit dem Zeigen ist der Vorgang der Selektion jedoch noch nicht abgeschlossen. Benutzer und System müssen die Selektion gegenseitig bestätigen. Das System tut dies durch entsprechende visuelle Hervorhebung (Bounding Box, Wireframe-Darstel-

lung, farbliche Markierung, etc.), auditive Meldung oder im Falle des Vorhandenseins von mit Force Feedback ausgestatteten Eingabegeräten wie Datenhandschuhen auch durch auf den Körper übertragene Kräfte oder Vibrationen. Der Benutzer kann die Selektion ebenfalls durch entsprechende Ereignisse wie Sprachbefehle, Gesten oder Drücken eines Knopfes bekräftigen.

Eine nahtlose Veknüpfung zwischen direkten und indirekten Methoden bildet die egozentrische Go-Go Interaktionstechnik [Pou 96]. Sie baut auf dem Konzept der Virtual Hand auf. Innerhalb von $2/3$ der Armlänge können virtuelle Objekte direkt selektiert und manipuliert werden. Über diese Distanz hinaus erfolgt eine nicht-lineare Abbildung auf einen den Arm verlängernden Vektor, sodass auch weiter als eine Armlänge entfernte Objekte mit der Virtual Hand erreicht werden können. Das heißt, die Länge des Vektors kann vom Benutzer bestimmt werden. Virtuelle Objekte lassen sich dann indirekt in der Entfernung manipulieren oder für eine direkte Manipulation zum Betrachter heranholen.

Komplexe Manipulationen von Objekten basieren meist auf zu Grunde liegenden elementaren Manipulationen. Eine Klasse bilden affine Transformationen wie Translation, Rotation und Skalierung. Sie ermöglichen zum Beispiel Kompositionsszenarien, in denen komplexe Objekte aus Primitivobjekten zusammengesetzt werden. Dabei werden bevorzugt Bewegungen der Hand als Transformationen auf virtuelle Objekte abgebildet. Weitere formerhaltende Manipulationen bilden Farb- oder Texturänderungen.

Formänderungen eröffnen darüber hinaus viel mehr Möglichkeiten Manipulationen an Objekten vorzunehmen, können aber auf Grund ihrer mathematischen Komplexität nur teilweise in Echtzeit-Anwendungen implementiert werden.

2.3.6.3 Systemsteuerung

Die Systemsteuerung beschreibt das Absetzen von Kommandos, das Umschalten von Modi sowie das Einstellen von Parametern, die den Zustand oder das Verhalten des Systems ändern. Hierzu gehören zum Beispiel das Laden von Szenen oder

zusätzlicher Objekten, das Ändern physikalischer Gesetzmäßigkeiten (Schwerkraft, Kollisionserkennung) oder das Umschalten zwischen verschiedenen Rendering- (solid/wireframe, stereo/mono), Fortbewegungs- oder Manipulationsmodi. Eine Systemsteuerung kann mit unterschiedlichen Techniken realisiert werden.

Das Einblenden von zweidimensionalen Menüs im Sichtbereich des Benutzers ist eine davon. Durch Selektion von Menüpunkten können entsprechende Kommandos ausgelöst werden. Eine spezielle Art von Menüs sind die TULIP-Menüs (Three-Up, Labels In Palm), die mit der Virtual Hand Metapher verbunden werden. Dabei werden zur virtuellen Repräsentation der Hand an jedem Finger Schriftzüge eingeblendet, die Menüpunkten entsprechen. Trägt der Benutzer Pinch Gloves, löst eine Geste eine Aktion dem entsprechend angezeigten Menüpunkt aus.

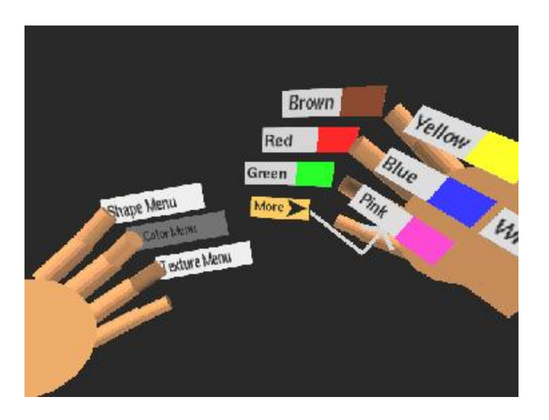


Abbildung 2-30: TULIP-Menü

Eine weitere beliebte Form der Systemsteuerung ist die Spracheingabe. Man unterscheidet dabei einfache, bei der einzelne Wörter oder Wortketten entsprechenden Kommandos zugeordnet werden und komplexeren Spracherkennung, die einen natürlichen Dialog mit dem System zulassen.

Gestische Kommandos beinhalten nicht nur starre Handhaltungen (Gesten als solche), sondern auch in Verbindung mit einer bestimmten Handhaltung durchgeführte

Bewegungen im Raum. So könnte ein mit einer Faust in die Luft gezeichnetes „C“ zum Beispiel eine Farbpalette erscheinen lassen.

Eine weitere Metapher zur Systemsteuerung bilden Werkzeuge. Sie funktionieren ähnlich den Werkzeugen in einem Bildverarbeitungsprogramm. Dort erzeugen zum Beispiel Pinsel und Airbrush unterschiedliche Linien. In VR-Anwendungen kann die Auswahl eines entsprechendes Werkzeuges eine Änderung des aktuellen Manipulations- oder Fortbewegungsmodus bewirken. Während Turnschuhe beispielsweise nur auf eine langsame Bewegung am Boden beschränkten, könnten Flügel eine schnellere Bewegung mit mehr Freiheitsgraden erlauben. Diese Werkzeuge könnten an einem virtuellen Werkzeuggürtel hängen und mittels Selektion wählbar sein oder physisch als Eingabegerät in Form von Wands vorhanden sein.

3 Das VR Objekt Display

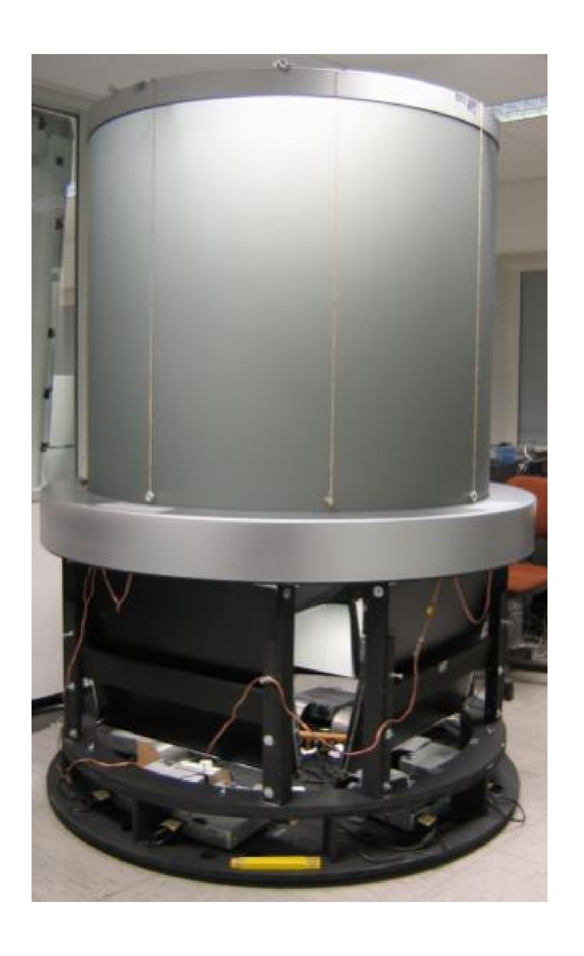


Abbildung 3-1: Prototyp VR Objekt Display

In klassischen VR-Umgebungen wie CAVE oder HMD befindet sich die virtuelle Welt um den Betrachter herum, er schaut wie durch Schaufenster auf virtuelle Objekte. Zudem schränken Installationen wie die CAVE den physischen Bewegungsfreiraum ein. Um Objekte aus unterschiedlichen Blickwinkeln zu sehen, müssen sie mithilfe von Eingabegeräten gedreht werden, was durch die Abstraktion eine gewisse Distanz zum Objekt schafft.

Das VR Objekt Display projiziert virtuelle Objekte innerhalb eines zylindrischen Projektionskörpers. Im Gegensatz zur CAVE-Technologie lässt das VR Objekt Display dem Betrachter alle physischen Freiheiten, er kann um ein virtuelles Objekt komplett

herumgehen, es aus jedem erdenklichen Blickwinkel erkunden wie es volumetrische Displays zulassen. Dies entspricht natürlichen Gewohnheiten und schafft eine starke Präsenz des dargestellten Objekts. Auf Grund seiner Dimensionen ist das VR Objekt Display ausgesprochen immersiv. Sogar lebensgroße Personen lassen sich darstellen. Die betrachterbezogene Darstellung kommt der eines Hologramms sehr nahe. Somit wird auch eine weiterführende Interaktion mit Objekten erleichtert.

Anwendungsszenarien Eine Reihe von Anwendungsbeispielen sollen die vielfältigen Einsatzmöglichkeiten des Displays umreißen. Auf Grund seiner Form eignet sich das VR Objekt Display hervorragend zur Darstellung einzelner virtueller Objekte.

Dies begünstigt vor allem Produktpräsentationen, in denen Kunden neue Produkte aus allen Perspektiven begutachten möchten. So zum Beispiel in der Modebranche. Der animierte Avatar eines Modells auf dem Catwalk könnte neueste Modeschöpfungen vorführen.

Im Bereich der Medizin kann das Display zur Ausbildung angehender Ärzte genutzt werden. Anhand eines lebensgroßen virtuellen Torsos können anatomische Grundlagen der menschlichen Physiologie vermittelt oder komplizierte Abläufe während einer Operation trainiert werden.

Im Telekommunikationssektor eröffnen sich neue Dimensionen der Telepräsenz. Ein 3D-Abbild des Gesprächspartners oder ganzer Gesprächsgruppen kann in voller Größe projiziert werden. In einer Kommunikation beispielsweise mit einer CAVE-Installation ließen sich alle in der CAVE befindlichen Personen filmen und im VR Objekt Display abbilden. Gleichzeitig könnte der Betrachter des VR Objekt Displays an einer entsprechenden Stelle der CAVE dargestellt werden.

Im musealen Umfeld kann das Display für Kunst und Archäologie Anwendung finden, um seltene Skulpturen und Artefakte einem breiteren Publikum zugänglich zu machen.

Für die Raumfahrt kann das Display zur Entwicklung und zum Test der korrekten Funktion von Satelliten eingesetzt werden.

3.1 Das Distributed Rendering System

Das von FIRST im Forschungsbereich ViSTA entwickelte Distributed Rendering System (DRS) ist eine VR-Software ähnlich CAVELib oder WorldToolKit, die auf Client-Server-Basis funktioniert und auf handelsüblicher Standardhardware lauffähig ist. Die Anwendungsebene bildet ein VRML-Browser, das von Holger Grahn entwickelte GL-View [GLV]. Das DRS ist in der Lage, 3D-Szenen zu laden und darzustellen.

Erweitert wird die Anwendungsebene durch eine Netzwerk- und Synchronisationsschicht. Damit ist es mit dem DRS möglich, Rendering-Cluster aufzubauen, wie sie für VR-Umgebungen wie CAVEs nötig sind. Dabei erlaubt das DRS mehrere Rendering-Clients pro Host-Rechner. Koordinierende Instanz im Cluster ist ein DRS-Server, der Änderungen am VRML-Szenengraphen an alle Rendering-Clients propagiert. Ein DRS-Cluster kann prinzipiell aus einer beliebigen Anzahl von Clients bestehen. Ein typisches Setup zeigt Abbildung 3-2.

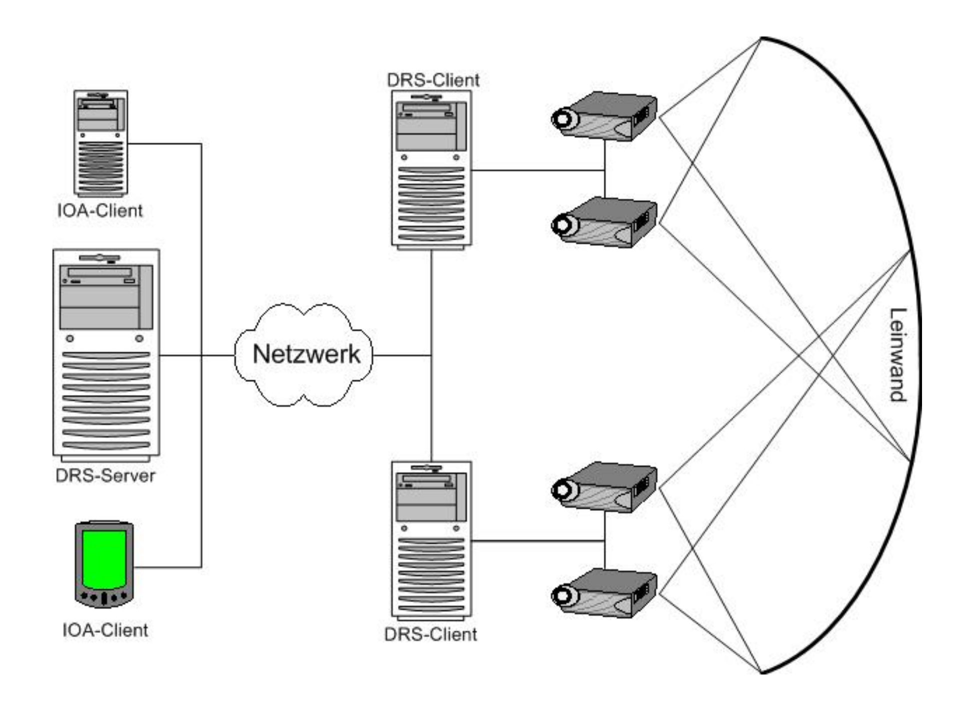


Abbildung 3-2: Typisches DRS-Setup

Die Konfiguration des Clusters ist über eine zentrale XML-Datei gelöst. Die Darstellung der von den Clients gerenderten Bilder kann am Bildschirm oder mittels Projektion auf Leinwände erfolgen. Dank eines von FIRST entwickelten automatischen Kalibrierverfahrens ist auch eine Projektion auf gekrümmte Oberflächen möglich.

Die Software ist in der Lage, stereoskopische Halbbilder zu erzeugen und unterstützt aktive wie passive Verfahren zur Bildtrennung.

Das DRS befindet sich in einem evolutionären Entwicklungsprozeß, um immer neuen Anforderungen gerecht zu werden. Eigens entwickelte Schnittstellen, auf die in Unterabschnitt 3.1.2 näher eingegangen wird, begünstigen diesen Entwicklungsprozeß. Sie sorgen dafür, dass auf neue Anforderungen schnell reagiert werden kann.

Das DRS ist auch die Software, mit der eine Darstellung am VR Objekt Display realisiert ist.

3.1.1 VRML

Wie im obigen Abschnitt angedeutet, ist VRML die Grundlage jeder mit dem DRS möglichen Darstellung. Da auch die im Rahmen dieser Arbeit entwickelten Interaktionskonzepte mit Techniken des VRML-Standards in Beziehung stehen, soll an dieser Stelle eine nähere Diskussion geführt werden.

Die *Virtual Reality Modeling Language* (VRML) ist ein internationaler Standard zum Austausch von 3D-Daten. VRML 2.0 wurde im April 1997 von der International Organization for Standardization (ISO) verabschiedet [VRML 97] und wird in der Namensregelung der ISO auch als VRML97 bezeichnet. Das Dateiformat basiert größtenteils auf dem bei Silicon Graphics Anfang der neunziger Jahre entwickelten Format für den Open Inventor. VRML integriert eine Reihe von anderen Standards wie zum Beispiel JPEG, um Objektoberflächen mithilfe von Texturen realistischer erscheinen zu lassen. Die Version 2.0 löste VRML 1.0 ab, das bis dahin noch einige Defizite aufwies. Beide Formate eignen sich hervorragend, um statische dreidimensionale Objekte zu speichern und auszutauschen. VRML97 bietet darüber hinaus aber noch an-

dere Möglichkeiten. So befähigt der Standard den Benutzer mit einer VR-Szene zu interagieren, Animationen zu erstellen und ein bestimmtes Verhalten der Szene über Programmierschnittstellen wie Java oder JavaScript zu definieren.

VRML beschreibt einen gerichteten azyklischen Szenengraphen, dessen Entitäten als Knoten bezeichnet werden. In dieser Hierarchie können Knoten andere Knoten enthalten, sie stehen also in einer Eltern-Kind-Beziehung zueinander. Es ist aber auch möglich, dass ein Knoten mehrere Eltern-Knoten besitzt. VRML definiert 54 unterschiedliche Knoten, darunter gruppierende Knoten wie der Group- oder der Transform-Knoten, geometrische Primitive wie Sphere oder Box, Knoten, die das Aussehen von Objekten definieren oder klangerzeugende (Sound-)Knoten. Die Daten oder Eigenschaften der Knoten werden in zugehörigen Feldern mittels unterschiedlicher Datentypen abgelegt.

Um Szenen einen interaktiven und lebendigen Charakter zu verleihen, beschreibt VRML einen speziellen Mechanismus, der eine Kommunikation zwischen den Knoten ermöglicht. Neben den normalen Feldern besitzen Knoten hierfür besondere Schnittstellen, die als ausgehende (eventOut) beziehungsweise eingehende (eventIn) Ereignisse bezeichnet werden. Eine sogenannte *Route* verknüpft ein ausgehendes mit einem eingehenden Ereignis gleichen Datentyps und stellt somit eine Verbindung zwischen zwei Knoten her. Auf diese Weise entstehen ganze Ereignisketten. Am Beginn jeder Ereigniskette stehen VRML-Sensoren. Sie bilden eine spezielle Klasse von Knoten und lassen sich in drei Gruppen unterteilen.

Zur Gruppe der *zeitgesteuerten Sensoren* gehört nur der TimeSensor. Oft durch einen TouchSensor (siehe unten) aktiviert, erzeugt er periodische Ereignisse innerhalb eines bestimmten Zeitintervalls. Verknüpft mit Interpolatoren, die wiederum mit einer sichbaren Geometrie verbunden sind, bildet der TimeSensor die Grundlage für Animationen.

ProximitySensor und VisibilitySensor bilden die Gruppe der *indirekt vom Benutzer gesteuerten Sensoren*. Beide erzeugen Ereignisse in Abhängigkeit von Position und Blickrichtung, die aus einer Bewegung des Benutzers innerhalb einer Szene resultie-

ren. Ein ProximitySensor könnte beispielsweise zur Simulation eines Bewegungsmelders verwendet werden. Sobald der Benutzer einen bestimmten Bereich betritt, wird eine Ereigniskette ausgelöst, an deren Ende zum Beispiel ein Licht eingeschaltet wird.

Die dritte und vielleicht wichtigste Gruppe begründen die *direkt vom Benutzer gesteuerten Sensoren*. Zu ihnen gehören TouchSensor, SphereSensor, CylinderSensor und PlaneSensor. Sie erzeugen Ereignisse infolge einer Eingabe, die der Benutzer mithilfe von Zeigegeräten tätigt. Üblicherweise kommen dabei Maus oder Trackball als Zeigegeräte zum Einsatz. Im Zuge der Betrachtung der Schnittstellen des DRS (siehe Abschnitt 3.1.2) wird sich jedoch herausstellen, dass auch Eingabegeräte mit einer dreidimensionalen Auflösung für Zeigeoperationen genutzt werden können.

Da Sensoren von konzeptioneller Natur sind, besitzen sie generell keine unmittelbare Ausprägung in visueller Form. Um direkt steuerbare Sensoren auszulösen, können sie mit sichtbaren Geometrien gruppiert werden. Zeigt der Benutzer auf eine mit einem direkt steuerbaren Sensor assoziierte Geometrie, kann im Falle des TouchSensors durch Drücken eines Knopfes am Zeigegerät eine Ereigniskette initiiert werden. Um komplexes Verhalten einer Szene zu definieren, werden in die Ereignisketten von TouchSensoren bevorzugt Script-Knoten eingearbeitet, die die Möglichkeit einer freien Programmierung bieten. Bei SphereSensor, CylinderSensor und PlaneSensor wird das Drücken des Knopfes mit einer Zugbewegung kombiniert (dragging). Die so ausgelösten Ereignisse beinhalten Rotationen (SphereSensor, CylinderSensor) beziehungsweise Translationen (PlaneSensor), die in der Regel direkt auf den übergeordneten Transform-Knoten geroutet werden, der ebenfalls die assoziierte Geometrie enthält. Das Ergebnis ist also eine entsprechende Transformation der Geometrie.

VRML gilt unterdessen als überholt. Der vom Web3D Konsortium entwickelte Nachfolger X3D bietet auf Grund seiner komponentenbasierten Architektur mehr Flexibilität als VRML97. Das auf XML basierende Format befindet sich derzeit kurz vor einer offiziellen Verabschiedung durch die ISO.

3.1.2 Schnittstellen

Das DRS bietet eine Reihe von Schnittstellen, die im Rahmen dieser Diplomarbeit genutzt wurden und an dieser Stelle näher erläutert werden sollen.

In Abschnitt 2.3.1.2 wird unter anderem auf das in einer CAVE bestehende Verzerrungsproblem hingewiesen. Bei der Entwicklung des DRS wurde eine entsprechende Schnittstelle, der sogenannte **Viewer**, zur Lösung dieses Problems definiert. Der Viewer repräsentiert die Position, für welche die Darstellung der Szene korrekt berechnet wird. Mithilfe eines Trackers kann diese in Echtzeit angepasst werden. Steht kein Tracker zur Verfügung kann der Viewer statisch in der Konfigurationsdatei des DRS festgelegt werden.

Bei einem CAVE-Aufbau befindet sich die Projektionsfläche um den Betrachter herum. Das Konzept des VR Objekt Displays sieht hingegen vor, dass der Benutzer die virtuelle Umgebung von außen sieht. Die Viewer-Schnittstelle und der damit verbundene Algorithmus zur Anpassung der Projektion sind so flexibel, dass auch für diesen Fall eine korrekte perspektivische Darstellung ermöglicht werden kann.

Das DRS bietet mit dem **3D-Cursor** eine Interaktionsschnittstelle zwischen Nutzer und virtueller Realität. Diese ermöglicht es mit unterschiedlichen Zeigegeräten VRML-Sensoren einer Szene anzusteuern. Beispielsweise ist es über das Auslösen eines TouchSensors möglich, eine Animation zu starten. Mithilfe eines SphereSensors lassen sich komplexe Objekte freihand rotieren, um bestimmte Details näher zu betrachten. Innerhalb der 3D-Szene wird der 3D-Cursor durch eine frei definierbare Geometrie visualisiert, die aus einer VRML-Datei geladen wird. Die einzelnen Interaktionsmodi (Fortbewegung, Selektion/Manipulation) können mit verschiedenen Formen des 3D-Cursors repräsentiert werden.

Die Steuerung des 3D-Cursors kann mittels klassischer Eingabegeräte wie der Maus erfolgen. Das DRS berechnet dabei ausgehend von der Orientierung der virtuellen Kamera- und Mausposition einen Richtungsvektor in die Szene zeigend. Einer-

seits kann der 3D-Cursor in einem festen Abstand auf diesem Richtungsvektor plaziert werden, andererseits kann er aber auch mittels Ray-Casting auf die Oberfläche des in Strahlrichtung liegenden Objektes versetzt werden. Der letzte Modus ist einfacher zu bedienen, da auch ohne stereoskopische Projektion erkennbar ist, über welchem VRML-Sensor sich der 3D-Cursor befindet.

Des Weiteren kann der 3D-Cursor direkt mit dreidimensionalen Eingabedaten manipuliert werden. Hierfür bietet sich die Verwendung von Tracking-Systemen an. Die Position aus dem Koordinatensystem des Trackers kann dabei direkt (isomorph) auf die Position des 3D-Cursors innerhalb der Szene übertragen werden. Darüber hinaus kann die Positionsberechnung des 3D-Cursors auch indirekt (nicht-isomorph) erfolgen. Der hierfür notwendige Richtungsvektor kann aus Kopf- und Handposition oder aus Handposition und ihrer Orientierung bestimmt werden. Die endgültige Position des 3D-Cursors wird wieder über einen festen Aktionsradius oder Ray-Casting berechnet.

Die **Input Output API** (IOA) wurde entwickelt um verschiedenste Ein- und Ausgabegeräte in einen DRS-Cluster einzubinden. Die flexible und leicht erweiterbare Schnittstelle ist prinzipiell in der Lage Ein-/Ausgabedaten jeglicher Art zu transportieren, um die Interaktion mit einem VR-Szenario und die Bedienung des DRS zu erleichtern.

Grundlage hierfür ist zunächst eine Client-Server-Architektur. Das standardisierte Protokoll der Schnittstelle basiert auf der TCP/IP-Protokollfamilie und setzt speziell auf dem UDP-Protokoll[8] auf. Hieraus lassen sich einige Vorteile ableiten. Durch die Verwendung von UDP wird eine optimale Performanz des Systems gewährleistet und Latenzen werden minimiert. TCP/IP als de facto Internetstandard spricht für eine Plattformunabhängigkeit. Sockets werden von jedem Betriebssystem unterstützt und bilden eine Art universellen Bus zwischen Client- und Serverkomponente. Außerdem kann das definierte Protokoll der Schnittstelle in vielen Programmiersprachen implementiert werden. All diese Umstände ermöglichen es, separate Anwendungen zu entwickeln, die über eine Instanz eines IOA-Clients mit dem DRS kommunizieren, in das gleich-

[8]UDP: User Datagram Protocol

zeitig auch ein IOA-Server integriert ist. Eine solche Applikation kann folglich auch auf einem entfernten Rechner ausgeführt werden, wo sie ein oder mehrere Eingabegeräte verarbeitet und dann entsprechende Signale zum DRS sendet. Diese massive Verteilung (siehe Abbildung 3-2) hat den Vorteil, dass das DRS von sämtlicher Verarbeitung von Eingabegräten entlastet werden kann, was sich in Performance-Gewinnen widerspiegelt. Außerdem können aus Sicht des Software Engineerings Ein- und Ausgabegeräte unabhängig vom DRS implementiert werden, was sich positiv in der Qualität der Software niederschlägt. Das heißt, es können Treiber für verschiedenste Ein- und Ausgabegeräte geschrieben werden, ohne dass ein stabiles DRS in seiner Entwicklung beeinträchtigt wird.

Die Input Output API entstand aus einer praktischen Semesterarbeit und wurde zunächst in C++ implementiert und zu einer Programmbibliothek kompiliert, die alle Funktionen eines IOA-Servers wie Clients (siehe Abbildung 3-3) bereitstellt. Bei der Anforderungsanalyse wurden dabei zunächst essentielle Funktionalitäten identifiziert, die die Schnittstelle bereitstellen sollte. Im Zuge der Weiterentwicklung wurde der Funktionsumfang jedoch beträchtlich erweitert. So ist es mit der IOA möglich ferngesteuert Szenen zu laden, in ihnen zu navigieren, VRML-Viewpoints auszulösen oder auch beliebige Daten analog zum VRML-Route-Konzept an VRML-Knoten zu senden. Die IOA Schnittstelle erlaubt somit die Steuerung aller in Abschnitt 2.3.6 angeführten Interaktionsaspekte.

Derzeit gibt es IOA-Treiber für Eingabegeräte wie Joystick, Maus, Keyboard, das PCdash2[9] oder einen Orientierungssensor (Gyroskop). Dass die Schnittstelle auch portabel ist, zeigt eine Implementierung für PDAs[10], die über drahtlose Technologien wie Bluetooth oder WLAN kommunizieren. Weiterhin wurde das IOA-Protokoll im Rahmen eines anderen Projekts in C# umgesetzt. Diese Implementierung wird derzeit am Fraunhofer Institut für Werkstoff- und Stahltechnik (IWS) genutzt, um eine Sprachsteuerung für das DRS zu realisieren.

[9] PCdash2 der Firma Saitek: Tablett mit 32 frei programmierbaren Tasten
[10] PDA: Personal Digital Assistent, Handheld Computer

Ähnlich anderen VR-Software-Systemen wie CAVELib, WorldToolKit, Open Inventor oder Java3D besitzt das DRS ebenfalls eine Schnittstelle, die einen Zugriff auf den VRML-Szenengraph zur Laufzeit erlaubt und somit die Möglichkeit eröffnet, Knoten direkt zu manipulieren. Die als **LaverCAVE API** bezeichnete Schnittstelle befindet sich jedoch vor einer massiven Umstrukturierung und wurde aus diesem Grund nicht für diese Arbeit herangezogen.

IOAClient

```
+connect(server:const char *,id:const char *): bool
+deregisterFeedbackHandler()
+deregisterMessageHandler()
+enableDebug(Debug:IOADebug *)
+getNewInstance(): IOAClient *
+init(): bool
+isConnected(): bool
+quit(): bool
+registerFeedbackHandler(handler:IOAFeedbackHandler *)
+registerMessageHandler(handler:IOAMessageHandler *)
+send3DCursor(cursor:IOA3DCursorData *)
+sendMessage(target:const char *,message:const char *,length:long,requestAck:bool=false,timeout:long=5000): b
+sendViewer(viewer:IOAViewerData *)
+setCamTransform(transform:float [6])
+setCommand(cmdmask:long)
+setStringCommand(stringCommand:const char *)
```

Abbildung 3-3: Interface der Klasse IOAClient (bereitgestellte Methoden)

3.2 Versuchsaufbau

Während der Bearbeitung dieser Arbeit stand noch kein fertiger Prototyp des VR Objekt Displays zur Verfügung. Um ungeachtet dessen mit der Entwicklung eines Interaktionskonzepts und der damit verbundenen Software beginnnen zu können, wurde ein Versuchsaufbau eingerichtet, der in grundsätzlicher Beschaffenheit und im Visualisierungsverhalten dem intendierten VR Objekt Display entsprach. Hierfür wurden zunächst fünf Rechner mit entprechender Netzwerktechnik in einem 150 cm hohen und mobilen Regal untergebracht. Auf dem Regal, in relativer Augenhöhe, wurden vier 17 Zoll große TFT-Monitore quadratisch angeordnet, um in der Gesamtheit eine Darstellungsfläche von 360° zu suggerieren. Auf dieser Topologie aufbauend konnte

ein DRS-Mono-Setup (Abbildung 3-4) mit einem Server und vier Clients eingerichtet werden.



Abbildung 3-4: DRS-Mono-Setup des Versuchsaufbaus

An einem separaten Arbeitsplatz wurde die für das Interaktionskonzept notwendige Software entwickelt und im Zusammenspiel mit dem DRS-Mono-Setup getestet. Dafür war es notwendig Arbeitsplatz und DRS-Mono-Setup durch FastEthernet miteinander

zu verbinden, um einen zügigen Datentransfer sicherzustellen. Des Weiteren gehörten unterschiedliche im Rahmen der Entwicklung des Interaktionskonzepts zu testende Eingabegeräte zum Versuchsaufbau. Zu diesen gehörten unter anderem drei unterschiedliche Modelle von Polhemus Trackingsystemen, ein Datenhandschuh, ein PDA und eine Gyromaus. Die Kontrolleinheit des jeweiligen Polhemus Trackers und der Datenhandschuh wurden dabei an den separaten Arbeitsplatz über RS232-Schnittstellen angeschlossen. Der PDA integrierte sich als mobiles Endgerät über drahtlosen Technologien wie WirelessLAN IEEE 802.11 oder Bluetooth in die Netzwerktopologie.

Um den Sender der elektromagnetischen Tracking-Hardware mit in den Versuchsaufbau zu integrieren, wurde dieser mithilfe eines Stativs oberhalb der Bildschirme positioniert. Dadurch wurde die Gefahr von Interferenzen und Störung durch metallische Elemente der darunterliegenden Hardware verringert.

3.3 Aufbau des VR Objekt Displays

Das VR Objekt Display ist ein zylindrischer Körper mit einer Gesamthöhe von 2,10 m und einem Durchmesser von 1,50 m und besteht aus einem Sockel und einer Projektionsfläche. Der Sockel erstreckt sich vom Boden bis zu einer Höhe von einem Meter. Er enthält Stromversorgung, acht Projektoren, vier Spiegel sowie ein Entlüftungssystem. Die Projektoren sind paarweise im Rechteck angeordnet und jeweils mit entsprechenden Infitec-Filtern (siehe Abschnitt 2.2.3) zur Bildtrennung ausgestattet. Jeweils ein Spiegel befindet sich gegenüber einem Projektorenpaar. Über dem Sockel befindet sich der Projektionskörper, der aus einer zu einem Zylinder geschwungenen Rückprojektionsfläche besteht. In ausgerolltem Zustand sind deren Abmessungen 4,10 m x 1,20 m. In der daraus resultierenden vertikalen Naht verläuft eine Kabelführung zum Polhemus Long Ranger (siehe Abschnitt 3.5). Dieser ist an einem Kreuz befestigt, das das obere Ende des Projektionskörpers stabilisiert. Der Polhemus Long Ranger hat einen Durchmesser von 46 cm und ist so angeordnet, dass er die Darstellung (durch Schattenwurf) nicht behindert. Der Projektionskörper ist im Durchmesser geringfügig kleiner als der Sockel. Er besitzt einen Radius von 65 cm.

Daraus ergibt sich ein Absatz zwischen Sockel und Projektionsfläche, in dem sich vier von einem Surround-System angesteuerte Lautsprecher befinden. Abbildung 3-5 zeigt den schematischen Aufbau des VR Objekt Displays.

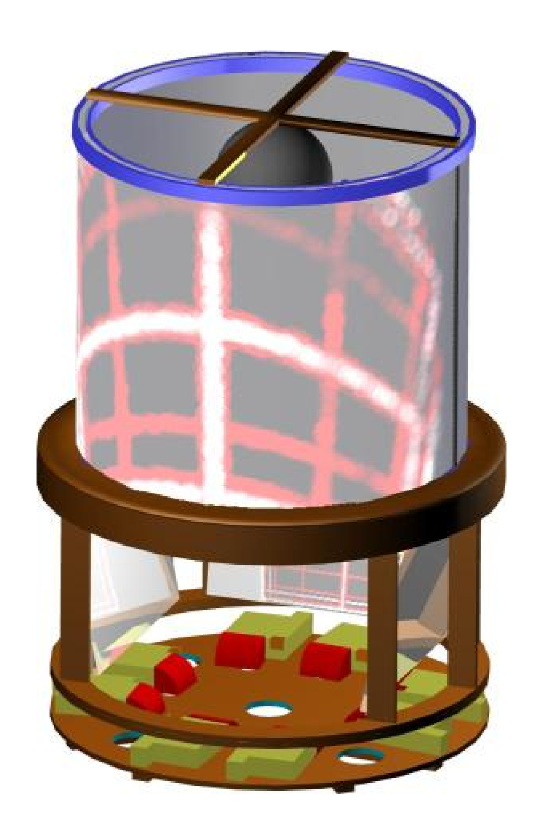


Abbildung 3-5: Aufbau VR Objekt Display

Des Weiteren besteht der Aufbau aus einem Cluster von fünf Rechnern, welche aus technischen Gründen keinen Platz mehr im Gehäuse des VR Objekt Displays fanden. Durch einen Strang von acht VGA-Kabeln, der aus dem VR Objekt Display führt, sind die Projektoren mit vier der Rechner verbunden. Dual Head Grafikkarten in diesen Rechnern übernehmen dabei jeweils die Ansteuerung von zwei Projektoren. Der fünfte Rechner ist DRS-Server. Er übernimmt zudem das Verarbeiten der Eingabegeräte und die Klangausgabe an ein Surround-Sound-System (Abbildung 3-6).

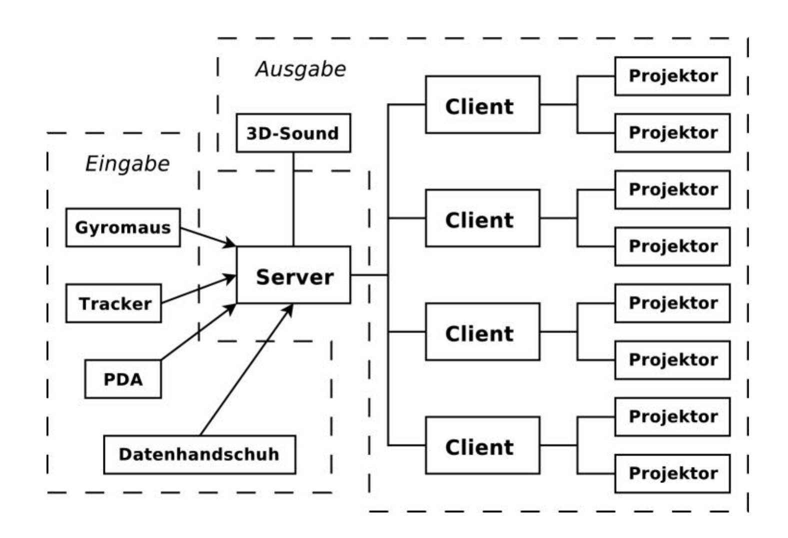


Abbildung 3-6: Schema VR Objekt Display

3.4 Funktionsweise

Da die Projektoren am Boden des Sockels schräg angestellt sind, fällt ihr Licht auf die gegenüber angeordneten Spiegel. Diese sind im Neigungswinkel justierbar und lenken das Licht auf die darüberliegende Projektionsfläche (siehe Abbildung 3-7). Ein Projektor leuchtet etwa 115° des Projektionskörpers aus. Dadurch überlappen sich Teile der Projektionen. Damit ein harmonisches Gesamtbild entsteht, müssen die Projektionen sanft ineinander überblendet werden. Zu diesem Zweck werden Blending Masken angewendet.

Auf Grund der Spiegelung und der gekrümmten Projektionsfläche entsteht ein verzerrtes Bild. Um diese Verzerrung auszugleichen, kommt ein eigens entwickeltes Kalibrierungsverfahren zum Einsatz. Ein spezieller Algorithmus berechnet dabei zunächst eine die gewölbte Projektionsfläche repräsentierende Geometrie. Diese dient bei der Echtzeitvisualisierung als Grundlage für ein 2-Pass-Rendering Verfahren, das im DRS implementiert ist. Im ersten Schritt wird das gerenderte Bild der Szene, anstatt es direkt in den Bildspeicher der Grafikkarte zu schreiben, zunächst in einem Zwischenspeicher abgelegt. Im zweiten Schritt wird dieses Bild als Textur auf die aus dem Ka-

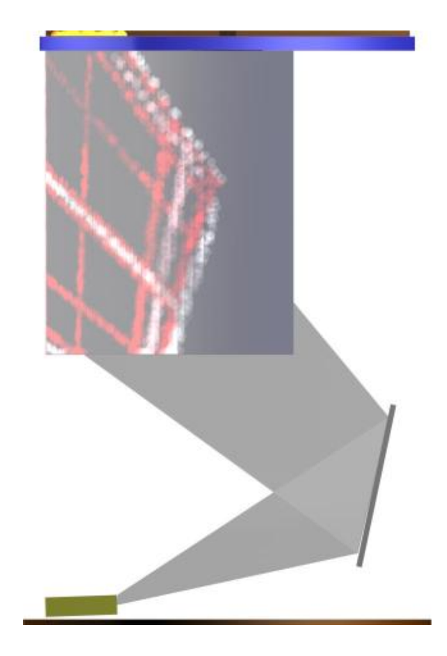


Abbildung 3-7: Strahlengang VR Objekt Display

librierungsverfahren hervorgegangene Geometrie angewendet. Das damit wiederum entstehende Bild wird jetzt ausgegeben und erscheint auf der geschwungenen Projektionsfläche unverzerrt.

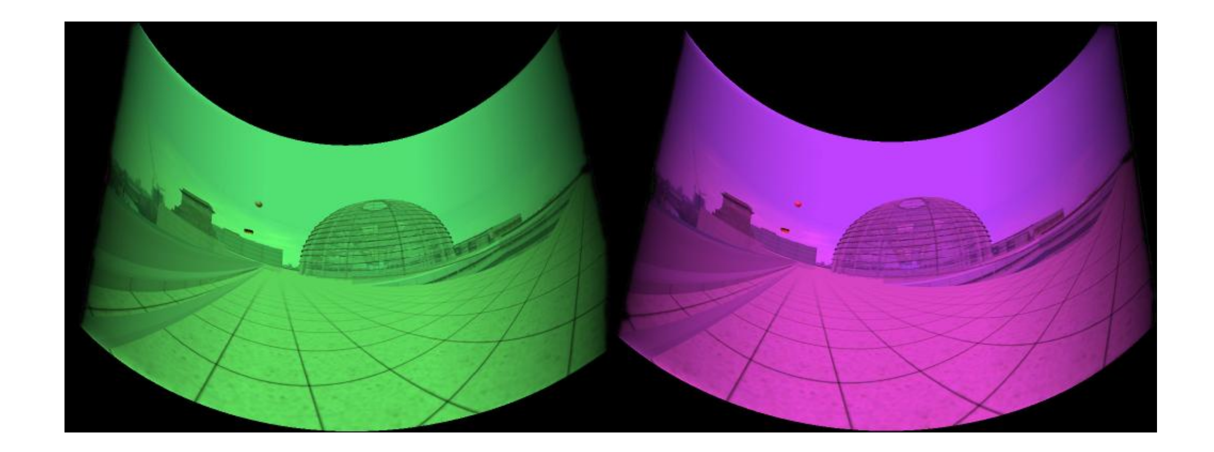


Abbildung 3-8: Entzerrung, Blending und Farbtransformation des VR Objekt Displays

Da heutige polarisationserhaltende Projektionsflächen nicht über einen geeigneten Biegeradius verfügen, kann die Polarisaton, als bewährte Methode stereoskopische Halbbilder voneinander zu trennen, für das VR Objekt Display nicht angewendet werden. Aus diesem Grund kommt beim VR Objekt Display für die stereoskopische Projektion das Infitec-Verfahren zum Einsatz. Infitec-Filter bewirken jedoch eine in Abschnitt 2.2.3 beschriebene Farbverzerrung. Für das VR Objekt Display wurde daher ein Pixelshader entwickelt, der eine Farbtransformation auf dem linken beziehungsweise rechten stereoskopischen Halbbild durchführt.

3.5 Verwendete Interaktionsgeräte

Abschnitt 2.3.2 sowie 2.3.3 stellen Eingabegeräte vor, die häufig in Verbindung mit VR-Systemen verwendet werden. Der folgende Abschnitt beschreibt eine Reihe von Eingabegeräten, die für eine Interaktion speziell mit dem VR Objekt Display Anwendung fanden.

Polhemus Tracker Die Geräte der Firma Polhemus basieren auf elektromagnetischem Tracking (siehe Abschnitt 2.3.2.4) und verfügen über sechs Freiheitsgrade. Für diese Arbeit standen drei Modelle zur Verfügung, die sich vor allem in ihrem Preis-Leistungs-Verhältnis unterschieden. Zwei ältere Geräte vom Typ ISOTRAK II und FASTRAK sowie ein neueres Gerät vom Typ PATRIOT. In der Grundausstattung bestehen alle Geräte aus einer Basiseinheit, einem Standardsender und ein oder mehreren Empfängern (siehe Abbildung 3-11, 3-9, 3-10). Sender und Empfänger sind über Kabel mit der Basiseinheit verbunden, die wiederum über die RS232-Schnittstelle an den Rechner angeschlossen wird. ISOTRAK II und PATRIOT stellen Einstiegsmodelle dar, welche über geringere Reichweite, Genauigkeit und Auflösung als der FASTRAK verfügen. Außerdem lassen sich an den FASTRAK bis zu vier Empfänger anschließen. Auf Grund seiner Eigenschaften (siehe Tabelle 3-1) wurde der FASTRAK für den Einsatz am VR Objekt Display favorisiert.

Neben den Standardkomponenten bietet Polhemus für seine Geräte eine Reihe

Eigenschaft	**Wert**
Reichweite	76 cm (mit garantierter Genauigkeit), bis 305 cm (mit abnehmender Genauigkeit)
Genauigkeit	0,08 cm beziehungsweise 0,15°
Auflösung	0,0005 cm beziehungsweise 0,025°
Verzögerung	4 ms
Aktualisierungsrate	120 Hz für einen Empfänger

Tabelle 3-1: Spezifikationen Polhemus FASTRAK

von Zubehörteilen an. Darunter zum Beispiel der *Poolball*, eine ausgehöhlte Billardkugel, in deren Inneren sich ein Empfänger verbirgt. Auf seiner Oberfläche befinden sich zudem zwei Druckknöpfe. Als Digitalisierwerkzeug gedacht, verfügt auch der sogenannte *Stylus* über einen Druckknopf. Poolball und Stylus lassen sich als Wand (siehe Abschnitt 2.3.3.2) einsetzen.



Abbildung 3-9: Polhemus Standardsender



Abbildung 3-10: Polhemus Standardempfänger (an Baseball-Mütze angebracht)

Zur Erhöhung der Reichweite auf 4,5 m ohne Reduzierung der Genauigkeit kann der Polhemus FASTRAK mit dem besseren Sender *Long Ranger* (Abbildung 3-12) ausge-

stattet werden. Ein Stylus und der Polhemus Long Ranger standen für die Diplomarbeit ebenfalls zur Verfügung.



Abbildung 3-11: Basiseinheit des Polhemus FASTRAK



Abbildung 3-12: Polhemus Long Ranger

Datenhandschuh Der Data Glove 5 der Firma Fifth Dimension Technologies gehört zur Klasse der Bend-Sensing Gloves (Abschnitt 2.3.3.1). Er funktioniert mithilfe von Leuchtdioden und besitzt eine Glasfaserschlaufe pro Finger. Der Datenhandschuh wird über die RS232-Schnittstelle an den Rechner angeschlossen und wurde zur Positions- und Orientierungsbestimmung der Hand mit einem Polhemus Standardempfänger versehen.



Abbildung 3-13: 5DT Data Glove 5 (mit Polhemus Standardempfänger)

PDA PDAs (Personal Digital Assistant) sind Taschencomputer, die in einer Hand gehalten werden können. Auf Grund ihrer Größe verfügen sie über einen Touch-Screen und benötigen keine Tastatur, um Eingaben zu tätigen. Trotzdem besitzen sie einige wenige Tasten, die spezielle Funktionen übernehmen, so zum Beispiel ein Cursor-Kreuz.

Um größtmögliche Mobilität zu gewährleisten, sind moderne Geräte mit drahtlosen Netzwerktechnologien wie Bluetooth oder WLAN ausgerüstet.



Abbildung 3-14: Hewlett Packard iPAQ H4150 mit laufender Anwendung

Gyromaus Gyromäuse sind Adaptionen herkömmlicher Desktop-Mäuse. Sie haben ebenfalls zwei Freiheitsgrade und zwei oder mehrere Tasten. Die Geräte werden vom Benutzer in der Hand gehalten und generieren Daten aus Orientierungsänderungen im Raum. Gyromäuse gibt es in unterschiedlichen Ausführungen, mit Kabelverbindung sowie kabellos über Funk kommunizierend.



Abbildung 3-15: Gyration Gyromäuse

4 Interaktionskonzepte

Das VR Objekt Display soll dem Benutzer den Eindruck vermitteln, dass sich das dargestellte virtuelle Objekt fest innerhalb des Projektions-Zylinders befindet und er mit diesem interagieren kann. Die Entwicklung hierfür geeigneter Interaktionskonzepte wird von mehreren Rahmenbedingungen begleitet. Die Interaktionen sollen auf dem Interaktionskonzept von VRML aufbauen und die vom DRS zur Verfügung gestellten Schnittstellen nutzen.

4.1 Entwurf

4.1.1 Fortbewegung

Konzeptbedingt eignet sich das VR Objekt Display nur zur Darstellung einzelner Objekte, deren Standpunkt sich in der Mitte des Display-Körpers befindet. Eine Fortbewegung durch große virtuelle „Landschaften“ ist nicht notwendig. Hingegen kann sich der Betrachter physikalisch dem Display nähern beziehungsweise davon entfernen sowie es umrunden, um andere Sichten auf das Objekt einzunehmen. Gewünscht ist ein Verhalten, das dem volumetrischer Displays (Abschnitt 2.3.1.5) gleicht.

Damit das dargestellte virtuelle Objekt für den Betrachter auch bei unterschiedlichen Betrachtungspositionen scheinbar an derselben Stelle steht, muß mittels Tracking die Kopfposition des Betrachters (Betrachter-Tracking) bestimmt werden. Zu diesem Zweck wurde ein Tracking-Sensor an einer Mütze befestigt (Abbildung 3-10), die vom Benutzer auf dem Kopf getragen wird. Die Position wird dem DRS über die Viewer-Schnittstelle mitgeteilt. Eine Änderung der Betrachterposition bewirkt daraufhin, dass die Abbildung auf der Zylinderfläche entsprechend angepaßt wird (Abbildung 4-1).

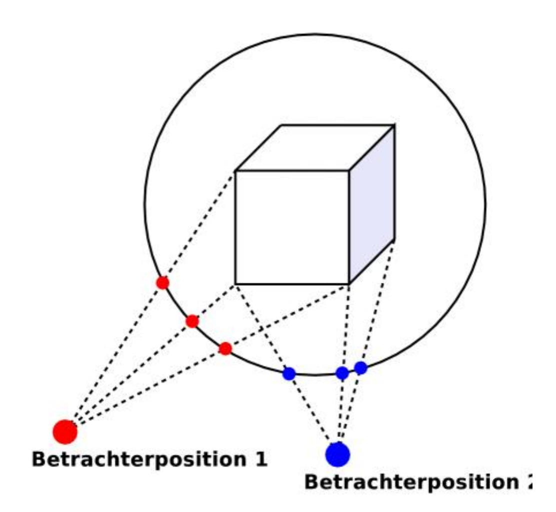


Abbildung 4-1: Betrachterbezogene Darstellung

4.1.2 Selektion und Manipulation

Da sich das virtuelle Objekt innerhalb des Projektionskörpers befindet, ist es physisch nicht direkt erreichbar. Isomorphe Methoden der Selektion beziehungsweise Manipulation können daher nicht angewendet werden. Demzufolge muss man auf indirekte Methoden ausweichen. Das DRS bietet hierzu mit der Schnittstelle zum 3D-Cursor eine entsprechende Unterstützung.

Zur Steuerung des 3D-Cursors ist die Erzeugung eines Richtungsvektors notwendig. Ausgehend von einer absoluten 3D-Position wird ein Ray-Casting entlang des Richtungsvektors in die virtuelle Szene durchgeführt. Schneidet der Strahl dabei das virtuelle Objekt, wird der 3D-Cursor an der Schnittposition angezeigt. Dies erfordert neben dem Tracking des Kopfes des Betrachters auch das Tracking der Hand beziehungsweise des in der Hand gehaltenen Eingabegerätes. Zur Generierung des Vektors werden vier unterschiedliche Möglichkeiten vorgeschlagen.

Die erste nutzt Position und Orientierung der Hand. Der Vektor beschreibt ausgehend von der Handposition eine Verlängerung des Unterarmes. Dieser Modus kann in Verbindung mit dem Datenhandschuh benutzt werden. Hierzu wird ein Tracking-Sensor wie in Abbildung 3-13 an der Hand angebracht.

Die zweite Methode basiert auf Tracking der Position und Orientierung des verwendeten Eingabegerätes. Die Methode ist mit der Benutzung einer Taschenlampe vergleichbar. Dadurch, dass das in der Hand gehaltene Eingabegerät und nicht die Hand selbst getrackt wird, ergibt sich zur Steuerung des Richtungsvektors mehr Bewegungsspielraum als bei der ersten Methode. Zu getrackten Eingabegeräten gehören Wands wie der Polhemus Stylus, der wie ein Stift in der Hand gehalten werden kann (Abbildung 4-3).

Bei der dritten Möglichkeit errechnet sich der Richtungsvektor aus Kopf- und Handposition, die Orientierungen der Tracking-Sensoren spielen dabei keine Rolle. Da die ersten beiden Methoden unabhängig von der Kopfposition sind, ermöglichen sie eine Positionierung des 3D-Cursors an für den Benutzer unsichtbaren Stellen. Die dritte Methode erfordert, dass der Benutzer die Hand auf Augenhöhe anhebt, um eine Selektion vorzunehmen.

Eine vierte Variante nutzt eine Gyromaus zur Erzeugung des Richtungsvektors. Sie verfügt über ein eigenes Tracking-System zur Orientierungsbestimmung. Als Ausgangspunkt für das Ray-Casting dient dabei die Kopfposition.

Die oben beschriebenen Steuerungen eines Richtungsvektors ermöglichen das Plazieren des 3D-Cursors auf einem virtuellen Objekt und somit seine Selektion. Zur Bestätigung der Selektion bieten sich visuelle Hervorhebungen an. So könnte das Objekt seine Farbe, Größe oder Beleuchtung ändern, selbst leuchten, blinken, eine zusätzliche Aura oder Bounding Box erhalten sowie in einer anderen Darstellungsart (wireframe, transparent) gezeichnet werden. Eine Hervorhebung könnte entweder automatisch vom DRS vorgenommen oder in der VRML-Szene implementiert sein.

Eine Manipulation von Objekten erfolgt durch das in VRML beschriebene Konzept der Sensoren. Der 3D-Cursor reagiert hierbei auf alle direkt vom Benutzer steuerbaren Sensoren (Abschnitt 3.1.1). Diese ermöglichen zum einen alle primitiven Transformationen. Sphere- beziehungsweise CylinderSensor erlauben eine Rotation und der Pla-

neSensor eine Translation von Objekten. Die Bewegung des 3D-Cursors wird dabei auf die entsprechende Transformation abgebildet. Zum anderen können komplexere in VRML implementierte Interaktionen mit dem TouchSensor ausgelöst werden. Das Auslösen der Sensoren geschieht durch ein konkretes Ereignis. Dieses kann zum Beispiel das Ausführen einer Handgeste oder das Drücken eines Knopfes sein.

Des Weiteren wäre es denkbar, zur Unterstützung der Interaktionsprozesses einen PDA einzusetzen. Ein Tracking des Gerätes, wie bei der oben beschriebenen zweiten Variante, würde zunächst eine Steuerung des 3D-Cursors erlauben. Auf dem PDA könnte dann eine Anzeige des Namens oder einer Beschreibung des aktuell selektierten Objektes erfolgen oder eine Liste der möglichen Animationen zur Auswahl angeboten werden. Die Manipulation von Objekten könnte durch den PDA zudem auf nicht-geometrische Objektattribute wie zum Beispiel die Materialeigenschaften (Farbe, Textur) ausgedehnt werden, ohne dass zusätzliche Widgets in der 3D-Darstellung eingeblendet werden müssten.

4.1.3 Systemsteuerung

Die Systemsteuerung am VR Objekt Display könnte wie in herkömmlichen Benutzeroberflächen mithilfe eines Menüs erfolgen, das direkt am unteren oder oberen Rand des Display-Zylinders angeordnet wird. Eine Umsetzung in VRML wäre denkbar. Dazu müssten jedoch neue VRML-Knoten entworfen werden, die entsprechende Kommandos im DRS umsetzen.

Die vorhandene Schnittstelle IOA (Abschnitt 3.1.2) des DRS bietet bereits vielfältige Möglichkeiten der Systemsteuerung an. Mithilfe entsprechender Gerätetreiber kann eine Reihe von Eingabegeräten zur Systemsteuerung eingesetzt werden. Knöpfe an Joysticks, Mäusen und Wands können bestimmten Kommandos zugeordnet werden. Auch von Datenhandschuhen erkannte Gesten der Hand können auf Systemkommandos abgebildet werden.

Ein vielseitigeres Eingabegerät zur Systemsteuerung ist der PDA. Die bei FIRST entwickelte Anwendung „DRSControl“ ist Implementierung eines IOA-Gerätetreibers

für den PDA. Die Anwendung ist in ihrer Erscheinung über eine XML-Datei konfigurierbar. Es können hierarchische Menüstrukturen aufgebaut werden. Ein einzelnes Menü wird am Touch-Screen angezeigt. Symbole bilden Menüpunkte, die je mit einem Systemkommando verknüpft sind. Es lassen sich Szenen laden, VRML-Viewpoints anwählen und Rendering-Modi umstellen (solid/wireframe, mono/stereo), um nur einige Funktionen zu nennen. Abbildung 4-2 zeigt einen Screenshot der Anwendung.



Abbildung 4-2: Screenshot DRSControl

Ein Tracking des PDA würde neben dem Interaktionsaspekt der Systemsteuerung gleichzeitig eine Selektion und Manipulation ermöglichen. Zusammen mit der Anzeige interaktionsbegleitender Informationen wäre der getrackte PDA das optimale Eingabegerät.

4.2 Evaluierung

Eine objektive Evaluierung der entwickelten Interaktionskonzepte auf Grundlage von umfangreichen Usability-Testreihen mit Probanden stehen zu diesem Zeitpunkt noch aus, da das VR Objekt Display noch nicht voll funktionstüchtig ist. An dieser Stelle kann daher nur eine subjektive Bewertung erfolgen. Diese basiert auf Erfahrungen, die beim Test der implementierten Interaktionskonzepte am Versuchsaufbau (Abschnitt 3.2) durch eine kleine Gruppe von Nutzern gewonnen wurden.

Fortbewegung Eine Begutachtung virtueller Objekte stellte sich dank Betrachter-Tracking als überaus intuitiv und Immersion fördernd heraus, da die physische Standpunktänderung der natürlichen Vorgehensweise zur Betrachtung von Objekten entspricht. Als störend erwies sich jedoch die Zeitverzögerung zwischen physischer Bewegung (besonders bei schnellen Bewegungen) und korrespondierender Darstellung, die auf interne Latenzen des Tracking-Systems und Datenverarbeitung zurückzuführen ist. Eine negative Auswirkung haben zudem metallische Objekte in der Umgebung. Sie stören das elektromagnetische Tracking. Die Folge ist eine Abweichung der perspektivischen Darstellung.

Selektion und Manipulation Bei der Steuerung des 3D-Cursors wurden unterschiedliche Beobachtungen gemacht. Wird der Richtungsvektor aus der Position und Orientierung nur eines Tracking-Sensors wie zum Beispiel beim Tracking des Eingabegerätes (Stylus) oder der Hand generiert, erfordert dies eine gewisse feinmotorische Geschicklichkeit, besonders wenn kleinere Objekte mit dem 3D-Cursor erfasst werden sollen. Dies stellte sich ebenfalls bei der Steuerung mit einer Gyromaus heraus, die zudem eine weitere Schwachstelle besitzt. Mit fortschreitender Nutzung akkumulieren sich die Fehler des gyroskopischen Tracking, die in einer starken Abweichung der gelieferten Orientierungsdaten resultieren. Die Gyromaus muss daher regelmäßig über die Betätigung eines separaten Knopfes zurückgesetzt werden.

Eine exakte Positionierung des 3D-Cursors wird außerdem durch die schon beim Betrachter-Tracking beschriebene Zeitverzögerung beeinträchtigt. Ferner hat sich gezeigt, dass eine visuelle Repräsentation des Zeigevektors hilfreich wäre, sie jedoch fehlt.

Wird der Richtungsvektor aus Kopf- und Handposition bestimmt, gestaltet sich die exakte Platzierung des 3D-Cursors einfacher. Jedoch muss hierfür die Hand ständig auf Augenhöhe gehalten werden, was nach einiger Zeit zu Ermüdungserscheinungen des Armes führt. Vom Standpunkt der Ergonomie aus gesehen ist dies nicht akzeptabel.

Bei der Manipulation von Objekten mittels Primitivtransformationen hat sich gezeigt, dass die im DRS implementierten Sensoren, vor allem die des SphereSensors, nicht intuitiv genug sind. Zur Rotation des Objekts muss nicht, wie man vielleicht erwartet, die Hand oder das getrackte Eingabegerät rotiert werden. Stattdessen muss eine Translation durchgeführt werden, die vom DRS auf eine Rotation abgebildet wird. Zur Aktivierung der Sensoren haben sich vom Datenhandschuh erkannte Gesten der Hand, vor allem eine Greifbewegung zur Faust, als sehr intuitiv herausgestellt (Abbildung 4-4). Jedoch wurde hierbei ein Seiteneffekt beobachtet. Das Formen einer Handgeste hat indirekten Einfluss auf die Orientierung des Tracking-Sensors, was zu einem „Verrutschen“ des 3D-Cursors führt. Bei der Betätigung des Knopfes am Polhemus Stylus (Abbildung 4-3) oder an einer Gyromaus zur Aktivierung von Sensoren trat der beschriebene Effekt zwar auch auf, aber nicht in einem Ausmaß wie bei der Benutzung des Datenhandschuhs.



Abbildung 4-3: Polhemus Stylus als Eingabegerät



Abbildung 4-4: Datenhandschuh als Eingabegerät

Systemsteuerung Ein PDA wurde anfangs vorgesehen, um Informationen über das selektierte Objekt anzuzeigen und den Interaktionsprozess zu unterstützen. Dazu sollte er mit einem Tracking-Sensor zur Steuerung des 3D-Cursors versehen werden. Es stellte sich jedoch heraus, dass das den PDA umgebende Feld oder metallische Bauelemente zu einer starken Fluktuation in den gemessenen Positionsdaten des elektromagnetischen Trackers führt. Dieser vielversprechende Ansatz musste daher verworfen werden.

Der ungetrackte PDA mit der Anwendung „DRSControl" wird dennoch zur Systemsteuerung eingesetzt, da die Bedienung auch für unerfahrene Benutzer verständlich ist.

4.3 Betrachtung zum Mehrbenutzerbetrieb

Alle bisherigen Überlegungen zielen auf eine Nutzung des VR Objekt Displays durch nur einen Benutzer ab. Ein einzelner Betrachter sieht jedoch stets weniger als 180° des Display-Zylinders. Es wäre daher möglich, das Display mehreren Benutzern zugänglich zu machen. Abbildung 4-5, Gleichung 1 sowie deren Umformungen Gleichung 2 und 3 zeigen den Zusammenhang zwischen der Entfernung des Betrachters zum Display und der Ausnutzung des Displays in Grad. Der Radius des Displays beträgt $r = 65cm$. 2α entspricht dem Winkel der sichtbaren Fläche bei einem Abstand a zum Display.

Würde man das Display auf drei Benutzer aufteilen wollen ($2\alpha = 120°$), ergibt sich ein Abstand $a = 65cm$ zum Display. Dieser wird aber als zu klein erachtet, da er ein besonders unnatürliches Verhältnis zwischen Konvergenz und Akkomodation (Abschnitt 2.2.3) und damit eine starke Anstrengung der Augen zur Folge hätte. Eine Betrachterentfernung von $a = 150cm$ wird hingegen als ausreichend groß angesehen. Somit ergäbe sich eine Ausnutzung des Displays von $2\alpha = 144,8°$. Daraus folgt, dass nur zwei Benutzer gleichzeitig am Display arbeiten könnten. Die Benutzer wären jedoch in ihrem Bewegungsspielraum eingeschränkt, was einen Bruch des Grundprinzips des VR Objekt Displays, ein virtuelles Objekt komplett umrunden zu können,

bedeutete.

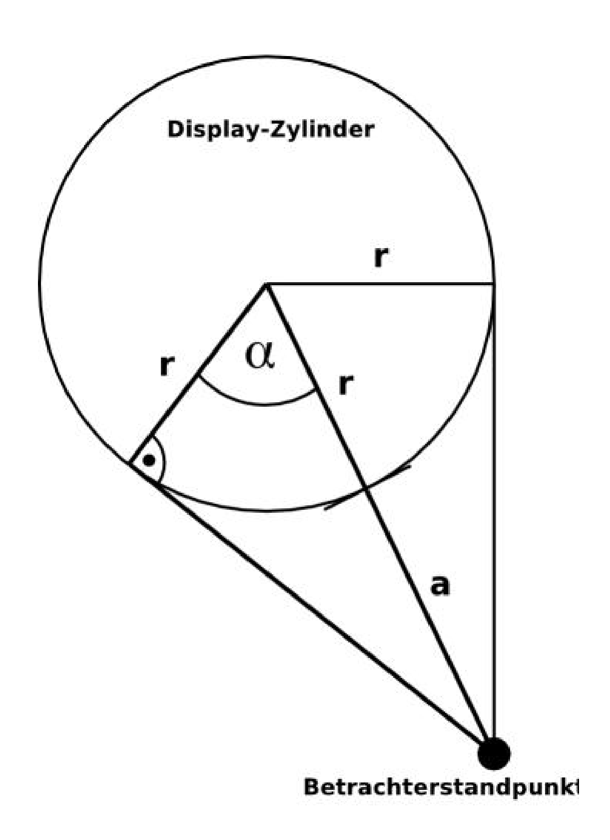


Abbildung 4-5: Entfernung und Displayausnutzung

$$\cos\alpha = \frac{r}{r+a} \tag{1}$$

$$a = \frac{r}{\cos\alpha} - r \tag{2}$$

$$\alpha = \arccos\left(\frac{r}{r+a}\right) \tag{3}$$

Des Weiteren müsste definiert werden, wie eine gemeinsame Arbeit an ein und demselben Objekt ermöglicht wird, da sich die beiden Teilnehmer nicht sehen können.

Jede Hemisphäre müsste die gleiche Hauptansicht des Objekts zeigen, sodass wenigstens eine gemeinsame Diskussionsgrundlage bestünde. Außerdem müsste der gemeinsame Zugriff auf das Objekt geregelt werden, um Deadlock-Situationen bei der Manipulation zu vermeiden.

Die Anwendung von Multiplex-Verfahren bei der Darstellung würde ein gemeinsames Betrachten eines Objekts ermöglichen, ohne dass das Display in Arbeitsbereiche aufgeteilt werden müsste. Das Infitec-Verfahren ist grundsätzlich in der Lage, mehr als zwei Bilder voneinander zu trennen. Dafür würden jedoch mehr Projektoren benötigt, die aus Gründen des Aufbaus aber nicht mehr untergebracht werden können. Multiplexing würde zudem auch nicht das Deadlock-Problem lösen.

5 Softwareentwicklung

Zur Umsetzung der Interaktionskonzepte wurden IOA-Gerätetreiber für Polhemus Tracking-Systeme und den Datenhandschuh Data Glove 5 der Firma Fifth Dimension Technologies implementiert. Das folgende Kapitel beschreibt den Prozess der Softwareentwicklung für die beiden im Rahmen der Arbeit entwickelten Anwendungen. Des Weiteren erfolgt eine kurze Beschreibung der Anwendungen.

5.1 Problemdefinition und Anforderungsanalyse

Aus den entwickelten Interaktionskonzepten ergeben sich verschiedene Anforderungen an die zu implementierende Software. Zunächst soll die IOA um eine Anbindung an die DRS internen Schnittstellen 3D-Cursor und Viewer (siehe Abschnitt 3.1.2) erweitert werden. Dazu ist es notwendig, die Klassen von IOA-Server und IOA-Client um entsprechende Funktionalität zu erweitern. Dazu sollen zum Beispiel die Methoden *send3DCursor()* und *sendViewer()* von der IOAClient-Klasse implementiert werden (siehe Abbildung 3-3). Entsprechendes gilt ebenfalls für die IOAServer-Klasse. Des Weiteren muss dafür gesorgt werden, dass der im DRS integrierte IOA-Server diese Daten den DRS-Einstellungen entsprechend aufbereitet und an die beiden DRS-Schnittstellen 3D-Cursor und Viewer weiterleitet.

Die Software muss die Rohdaten der verschiedenen Polhemus Tracker auswerten können und dazu deren unterschiedliche, aus den Handbüchern [POL] entnehmbaren, Protokolle implementieren. Da der Tracker ein eigenes Koordinatensystem verwendet, müssen die Daten mittels Koordinatentransformation in das Koordinatensystem der VRML-Szene überführt werden. Eine dreidimensionale wie textuelle Visualisierung soll die getrackten Positionssensoren zur Verifikation der Einstellungen veranschaulichen.

Weiterhin muss die Software die Ansteuerung und Kontrolle des Datenhandschuhs übernehmen. Da der Handschuh die Erkennung verschiedener Gesten der Hand ermöglicht, müssen diese entsprechenden IOA-Kommandos zuordenbar sein. Hierzu

soll eine Zuordnungstabelle umgesetzt werden. Zur Kontrolle soll die aktuell ausgeführte Geste textuell angezeigt werden.

Über Instanzen der Klasse IOAClient sollen die aufbereiteten Daten an den IOA-Server (DRS) gesendet werden. Die Einstellungen der Eingabegeräte soll komfortabel über eine grafische Benutzerschnittstelle erfolgen. Alle Konfigurationsparameter sollen dabei über die Laufzeit des Programms hinaus erhalten bleiben.

Auf Grund des Entwicklungsstatus und um die Software einfacher wartbar und erweiterbar zu halten, ist es von Vorteil, zwei Anwendungen den Eingabegeräten entsprechend zu entwickeln. Dieser Ansatz passt perfekt in die Komponentenarchitektur der IOA. Jede der beiden Anwendung wird als seperater IOA-Client fungieren. Die beiden resultierenden Anwendungen sollen die Namen „PolhemusGUI“ und „Glove“ tragen. Während PolhemusGUI für die Verarbeitung der verschiedenen Tracker verantwortlich ist, soll Glove eine Verbindung zum Datenhandschuh herstellen.

5.2 Entwurf

Der Entwurf erfolgt objektorientiert. Dazu wird die aus den verschiedenen Anforderungen resultierende Funktionalität entsprechenden Klassen und ihren Methoden zugeordnet.

Beim Aufbau einer geeigneten Programmstruktur wird das aus dem Softwareengineering bewährte Model-View-Controller (MVC) Entwurfsmuster angewendet. Dieses sieht vor, die Anwendung in verschiedene Komponenten zu zerlegen, die unterschiedliche Aufgaben übernehmen und in der Lage sind, miteinander zu kommunizieren (siehe Abbildung 5-1). Basis der Anwendung ist das Datenmodell (Model), das eine Abstraktion des umzusetzenden Problems in Programmform bildet. Die Benutzerschnittstelle (View) ist die visuelle Repräsentation des Datenmodells. Darüberhinaus bietet die Benutzerschnittstelle Interaktionsmöglichkeiten mittels konventioneller Eingabegeräte (Maus, Tastatur). Ausgelöste Aktionen werden an die Anwendungssteuerungskomponente (Controller) weitergeleitet. Diese hat Zugriff auf das Datenmodell und kann entsprechend Daten manipulieren. Das Datenmodell informiert wiederum

die Benutzeroberfläche über Änderungen, welche in einer Aktualisierung der Darstellung resultiert. Diese spezielle Architektur erhöht die Wiederverwendbarkeit der einzelnen Komponenten und ermöglicht zudem für ein und dasselbe Datenmodell mehrere Benutzeroberflächen zu entwerfen.

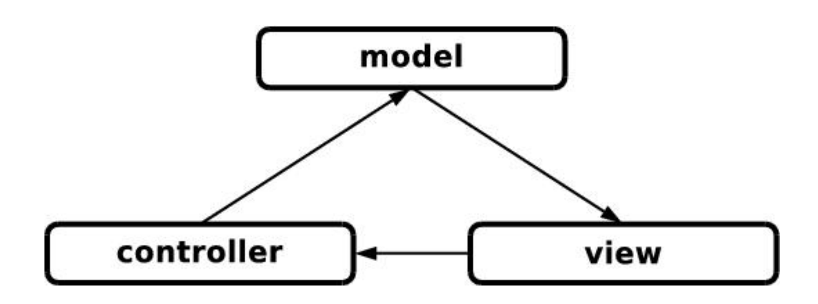


Abbildung 5-1: Model-View-Controller Modell

Bei modernen Implementierungen von Benutzeroberflächen wie Java Swing oder den Microsoft Foundation Classes (MFC) hat sich jedoch gezeigt, dass View- und Controller-Komponente eng miteinander verknüpft sind und eine Trennung nicht sinnvoll erscheint. Hieraus etablierte sich ein modifiziertes MVC-Modell (Abbildung 5-2), welches View und Controller zum User Interface vereint. Dieses Konzept wird auch bei den beiden im Rahmen der Diplomarbeit entwickelten Programmen angewendet.

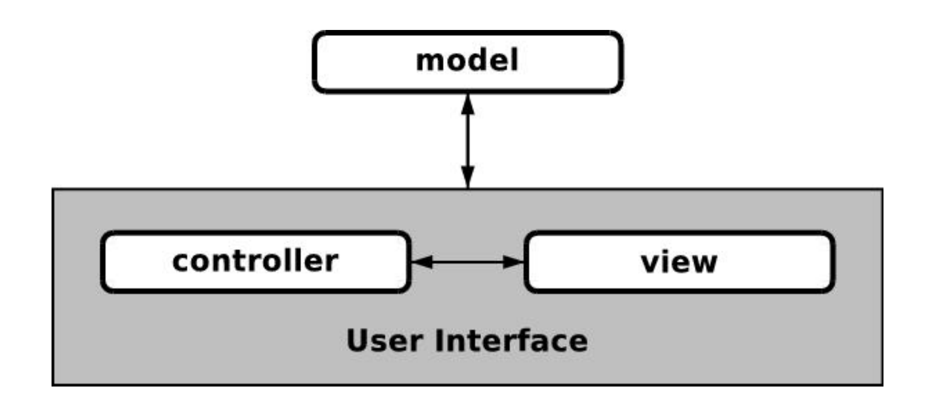


Abbildung 5-2: Modifiziertes MVC-Modell

Abbildung 5-3 und Abbildung 5-4 zeigen die Klassendiagramme der beiden Programme PolhemusGUI und Glove. Dabei sind die Klassen ihren, dem modifizierten MVC-Modell entsprechenden, Komponenten zugeordnet.

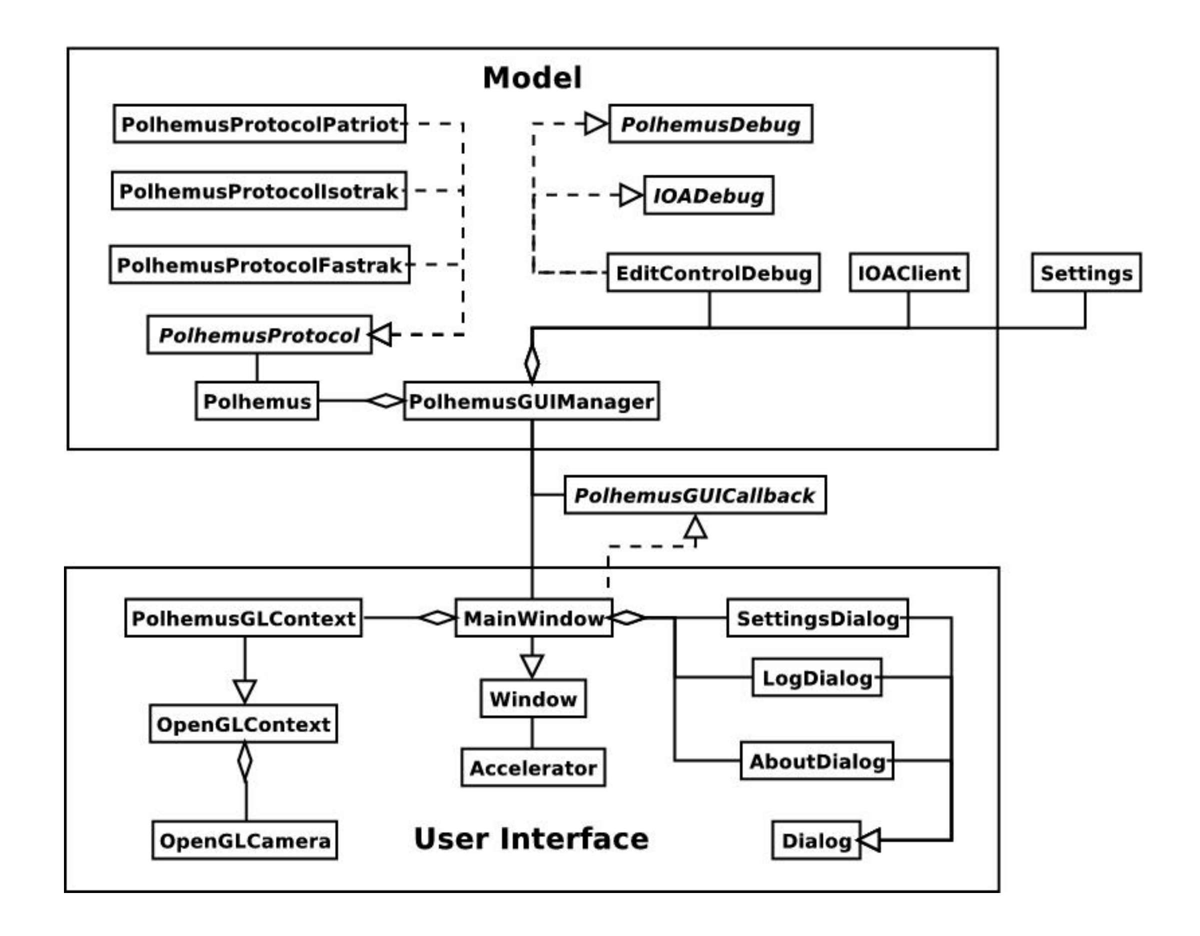


Abbildung 5-3: Klassendiagramm PolhemusGUI

Die Klasse Polhemus kapselt alle Funktionalitäten, um mit den Trackern zu kommunizieren. Dazu erzeugt sie eine Instanz der jeweils zu verwendenden Protokollklasse, in Abhängigkeit des benutzten Tracker-Modells. Kontrolliert wird die Klasse Polhemus durch die Klasse PolhemusGUIManager. Diese besitzt unter anderem auch eine Instanz eines IOAClients, die, wie bereits erwähnt, eine Kommunikation mit dem DRS sicherstellt. PolhemusGUIManager nimmt nicht nur eine zentrale Position im Daten-

modell ein, sondern bildet auch die äußere Schnittstelle zur Benutzerschnittstelle. Die Kommunikation zwischen Datenmodell und Benutzeroberfläche erfolgt bidirektional. Dazu hält MainWindow eine Referenz auf den PolhemusGUIManager und kann somit Methoden an ihm ausführen. Gleichzeitig implementiert sie die abstrakte Klasse PolhemusGUICallback, auf die der PolhemusGUIManager wiederum eine Referenz besitzt. Bei Datenänderung ist PolhemusGUIManager somit befähigt, über diesen Callback Nachrichten an das User Interface zu schicken. Die Klasse MainWindow ist eine Spezialisierung der Klasse Window und besteht einerseits aus PolhemusGLContext, der für eine dreidimensionale Visualisierung der Positionsdaten verantwortlich ist. Andererseit besitzt MainWindow verschiedene Spezialisierungen der Dialog-Klasse, welche unter anderem zur Anzeige von Programmeinstellungen oder Debugging-Informationen verwendet werden.

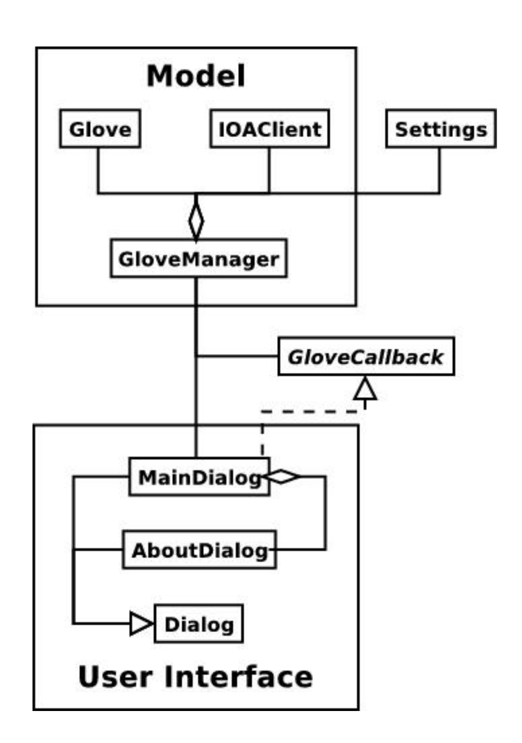


Abbildung 5-4: Klassendiagramm Glove

Die Klassenstruktur der Anwendung Glove ist sehr ähnlich der von PolhemusGUI. Die

Darstellung der Benutzerschnittstelle übernimmt hierbei die Klasse MainDialog. Die Kommunikation zwischen Benutzerschnittstelle und Datenmodell erfolgt analog der in PolhemusGUI. Die Klasse GloveManager ist wieder zentrales Kontrollorgan in der Model-Komponente. Sie enthält Instanzen der Klasse Glove und IOAClient. Die Klasse Glove bietet dabei Methoden an, um mit dem Datenhandschuh zu kommunizieren.

Wie in den Klassendiagrammen auffällig, besitzen die Managerklassen beider Anwendungen ein Objekt der Klasse Settings. Sie sorgt für eine persistente Datenhaltung der aktuellen Programmeinstellungen. Die Klasse nimmt im Diagramm eine gesonderte Position ein, da sie keiner der Komponenten des MVC-Modells zugeordnet werden kann.

5.3 Implementierung

Für die Implementierung der Anwendungen sind strenge Rahmenbedingungen gesteckt. Da es sich bei den zu entwickelnden Anwendungen um Erweiterungen des DRS handelt, liegt es nahe, dessen Systemplattform, Entwicklungsumgebung und Programmiersprache als Vorgabe zu verwenden. Aus diesem Grund sollen die Anwendungen ebenfalls für das Windows Betriebssystem entwickelt werden. Dazu bietet Microsoft komfortable Entwicklungsumgebungen aus der Visual Studio Familie an. Die Quellen von DRS, IOA und vielen verwendeten Bibliotheken liegen in C/C++ vor. Zudem ist es mit C++ möglich, das objektorientierte Paradigma umzusetzen.

Als Entwicklungsumgebung wurde Microsoft Visual C++ 6.0 gewählt. Zu den verwendeten Bibliotheken und Schnittstellen gehören neben der IOA die Windows API, FGlove und OpenGL sowie deren Erweiterung GLUT. Die in einer Bibliothek vorliegende Input Output API (IOA) dient der Kommunikation mit dem DRS und wird beim Kompilieren zum Programm hinzugelinkt. Die Windows API wird vorrangig verwendet, um Benutzeroberflächen zu erzeugen. Sie stellt darüber hinaus aber auch Funktionen bereit, welche die Kommunikation mit Peripheriegeräten an der RS232-Schnittstelle und eine parallele Programmierung mittels Threads ermöglichen. Zur dreidimensionalen Visualisierung der vom Tracker gelieferten Positionen kamen OpenGL und GLUT

zum Einsatz. FGlove ist die vom Hersteller Fifth Demension Technologies mitgelieferte Bibliothek zur Ansteuerung des Data Glove 5 Datenhandschuhs. Ihre Funktionen liefern die erkannten Gesten der Hand.

Anhang A enthält vollständige Klassendiagramme der beiden entwickelten Anwendungen.

5.4 Programmkurzbeschreibung

Abbildung 5-5 zeigt die Benutzeroberfläche der Anwendung PolhemusGUI. Zu sehen ist die dreidimensionale Darstellung eines Koordinatensystems, die zur besseren Anschauung um den Koordinatenursprung rotiert werden kann. Die Länge einer Koordinatenachse entspricht dabei einer virtuellen Längeneinheit. Zwei gelbe Kugeln repräsentieren die Positionen der Tracking-Sensoren nach einer Transformation in das VRML-Koordinatensystem. Eine textuelle Darstellung in der linken unteren Ecke des Fensters gibt die exakten Koordinaten der Sensoren wieder.

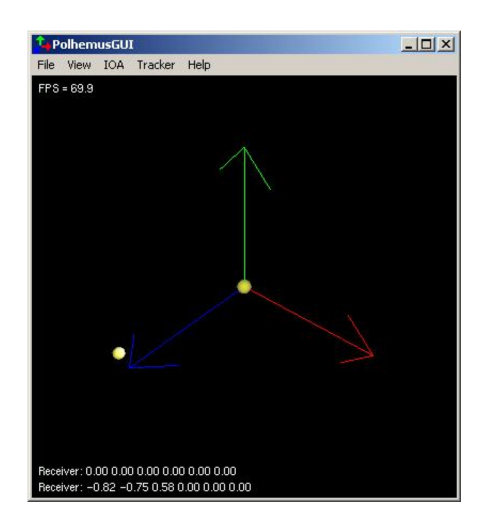


Abbildung 5-5: Screenshot der Anwendung PolhemusGUI

Zur Konfiguration der Anwendung kann über das Menü ein separater Dialog (Abbildung 5-6) aufgerufen werden. Im oberen Gruppenfeld (IOA) können IOA spezifische

Einstellungen vorgenommen werden. So lassen sich über Eingabefelder die Netzwerkadresse des IOA-Servers (DRS) und ein Identifikator für den IOA-Client angeben. Das untere Gruppenfeld (Polhemus) des Dialogs enthält Steuerelemente, die alle das Tracking-System betreffenden Einstellungen ermöglichen. Zunächst kann über ein Kombinationsfeld das angeschlossene Polhemus Tracking-System und das damit zu verwendende Protokoll definiert werden. Zur Auswahl stehen „Isotrak II“, „Patriot“ und „Fastrak“. Mithilfe weiterer Steuerelemente lassen sich beispielsweise Anschlusseinstellungen konfigurieren, die Transformation der Positionsdaten aus dem Koordinatensystem des Trackers in das von VRML (OpenGL) beeinflussen oder die Parameter des im Tracking-System integrierten Filteralgorithmus justieren.

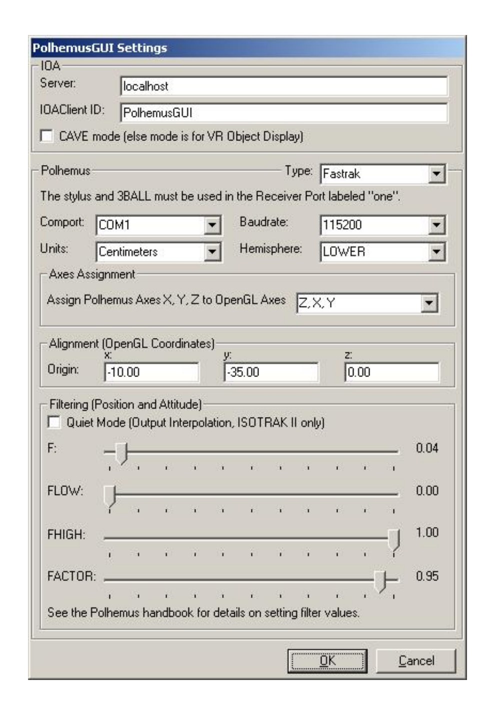


Abbildung 5-6: Screenshot PolhemusGUI Settings Dialog

Abbildung 5-7 zeigt die Benutzeroberfläche der Anwendung Glove. Im oberen Gruppenfeld (Gesture-Command Map) erfolgt zunächst die textuelle Ausgabe der aktuell vom Datenhandschuh erkannten Geste in einem statischen Textfeld. Darunter be-

finden sich zwei Kombinationsfelder, mit denen eine Zuordnung zwischen erkannter Handgeste und IOA-Kommando vorgenommen werden kann. Im ersten Kombinationsfeld (Gesture) wählt der Benutzer zunächst die gewünschte Geste. Existiert bereits ein zugeordnetes IOA-Kommando, ändert sich entsprechend der Inhalt des zweiten Kombinationsfeldes (IOA Command). Wählt man ein Kommando aus dem zweiten Kombinationsfeld, wird es automatisch der im ersten Kombinationsfeld spezifizierten Geste zugeordnet. Im unteren Gruppenfeld (Settings) können Einstellungen den Datenhandschuh und die IOA-Schnittstelle betreffend vorgenommen werden. Über ein Kombinationsfeld lässt sich die COM-Schnittstelle (RS232) angeben, über die der Datenhandschuh an den Rechner angeschlossen ist. Darunter befinden sich zwei Eingabefelder, in die die Netzwerkadresse des IOA-Servers und ein Identifikator für den IOA-Client eingetragen werden können.

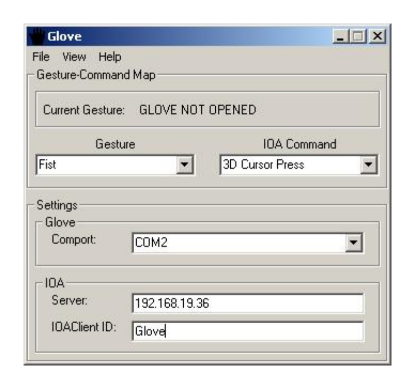


Abbildung 5-7: Screenshot der Anwendung Glove

Sind alle Einstellungen korrekt vorgenommen worden, kann über das Menü der beiden Anwendungen eine Verbindung zum entsprechenden Eingabegerät und zum DRS aufgebaut werden. Während des Betriebs lassen sich die Anwendungen minimieren und in die System-Ablage (System Tray) des Windows Desktops einfügen.

6 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurden verschiedene Interaktionskonzepte für das VR Objekt Display entwickelt, prototypisch umgesetzt und bewertet. Hierzu wurde zunächst eine Einteilung der Nutzerinteraktion hinsichtlich Fortbewegung, Selektion beziehungsweise Manipulation und Systemsteuerung vorgenommen.

Im Gegensatz zu etablierten VR-Umgebungen wie der CAVE erfolgt die Fortbewegung in der vom VR Objekt Display erzeugten virtuellen Welt auf keiner abstrakten Ebene. Stattdessen wird eine echte physische Standpunktänderung des Benutzers im Raum maßstabsgetreu auf eine Bewegung in der virtuellen Welt abgebildet. Ein ortsgebundenes virtuelles Objekt, innerhalb des VR Objekt Displays projiziert, kann auf natürliche Art und Weise vom Benutzer umgangen und so aus jedem Blickwinkel betrachtet werden. Diese Form der Fortbewegung erzeugt im Bewusstsein des Benutzers eine natürliche Präsenz des Objekts und spricht für den hohen Grad an Immersion, den das VR Objekt Display bietet. Möglich wird dies jedoch nur durch die Bestimmung der Kopfposition des Benutzers mittels eines elektromagnetischen Tracking-Systems. Die im Rahmen der Arbeit erstellte Anwendung PolhemusGUI verarbeitet hierfür die Daten des Tracking-Systems und leitet sie an die darstellungserzeugende Software (DRS) des VR Objekt Displays weiter. In der Praxis zeigen sich jedoch die Schwächen des elektromagnetischen Trackings. Einerseits erzeugt die zu hohe Latenz Simulatorschwindel. Andererseits führt die Anfälligkeit gegenüber anderen Strahlungsquellen und Metallteilen in der Konstruktion des VR Objekt Displays zu gravierenden Fehlern in der sonst recht genauen Positionsbestimmung der Polhemus Systeme. Die Folge ist eine vom Benutzer als störend wahrgenommene Abweichung in der betrachterbezogenen Darstellung.

Die durch das Betrachter-Tracking erschaffene Präsenz des virtuellen Objekts vereinfacht dessen Selektion und Manipulation. Leider ist das Objekt aufgrund der physischen Barriere des Projektionskörpers nicht direkt erreichbar. Eine Selektion und Manipulation erfolgt daher indirekt mithilfe eines 3D-Cursors. Hierfür übernimmt die Anwendung PolhemusGUI das Tracking eines weiteren Positionssensors. Die Selekti-

on von Objekten durch Eingabegeräte, die ähnlich einer Taschenlampe gehandhabt werden, eignet sich auch für unerfahrene Benutzer. Eine Manipulation von Objekten auf der Grundlage von VRML-Sensoren wurde als nicht intuitiv genug bewertet, da besonders Geometrie-Sensoren (Sphere-, Cylinder-, PlaneSensor) für die jeweilige Transformation eine nicht entsprechende Bewegung erfordern. Für das Auslösen von Sensoren haben sich indessen von einem Datenhandschuh erfasste Handgesten hervorgetan. Hierfür wurde die Anwendung Glove entwickelt. Ein für die Interaktion vielseitig einsetzbarer PDA ließ sich aufgrund von Störfeldern nicht mit dem elektromagnetischen Tracking kombinieren. Eine Manipulation weiterer Objekteigenschaften über dessen berührungsempfindlichen Bildschirm konnte daher nicht umgesetzt werden.

Der PDA wurde dennoch zur Systemsteuerung eingesetzt. Die vorhandene Anwendung „DRSControl“ wurde hierfür in ihrer Konfiguration dem VR Objekt Display angepasst.

6.1 Ausblick

Nach Fertigstellung des VR Objekt Displays sollten die im Rahmen der Arbeit entwickelten Interaktionskonzepte umfassenden Usability-Tests unterzogen werden, um eine wissenschaftlich fundierte Bewertung vornehmen zu können. Hierfür könnten im Vorfeld eine Reihe von Schwachstellen beseitigt werden, die beim Test am Versuchsaufbau entdeckt wurden.

Die auftretenden Abweichungen im Tracking aufgrund von statischen Störfeldern und metallischen Objekten könnten mithilfe eines Kalibrierungsverfahrens ausgeglichen werden. An einer repräsentativen Anzahl von Positionen könnten die vom Tracker gelieferten Daten mit der tatsächlichen Position verglichen werden. Das Resultat wäre ein Algorithmus, der anhand der vom Tracker gelieferten Werte eine Interpolation zwischen den bei der Kalibrierung bestimmten tatsächlichen Werten vornimmt. Auf diese Weise könnte ein angenähertes tatsächliches Ergebnis geliefert werden.

Zur Integration des PDAs in den Interaktionsprozess könnte ein anderes Tracking-

Verfahren (z.B. optisch) genutzt werden. Eine provisorische Lösung könnte auch eine leichtgewichtige Konstruktion bieten, welche für einen ausreichenden Abstand zwischen PDA und elektromagnetischem Tracking-Sensor sorgt.

Bei der Entwicklung der Interaktionskonzepte wurden weiterhin eine Reihe von Unzulänglichkeiten im DRS identifiziert. Zur Unterstützung der Selektion sollte der Richtungsvektor eine visuelle Repräsentation in Form eines Strahls erhalten. Zudem könnte eine Bestätigung der Selektion in entsprechender Art und Weise automatisch vom DRS vorgenommen werden. Weiterhin wäre eine Überarbeitung der VRML-Geometrie-Sensoren in der Form sinnvoll, dass Positionen und Orientierungen eines Tracking-Systems direkt auf den entsprechenden VRML-Sensor abgebildet werden können. Denkbar wäre auch die Implementierung eines „TrackerSensors“, der die gleichzeitige Manipulation von Position und Orientierung auf der Basis von Tracking-Daten erlaubt.

Literatur- und Quellenverzeichnis

[Bahr 96] BAHR, D.; LANGHANS, K.; et al.: *FELIX: A volumetric 3D laser display.* In: Projection Displays II. Proceedings of SPIE Vol. 2650, 1996, S. 265-273

[Bor 94] BORMANN, S.: *Virtuelle Realität. Genese und Evaluierung.* Addison-Wesley, 1994

[Bow 00] BOWMAN, Doug A.; KRUIJFF, E.; et al.: *3D User Interface design: Fundamental Techniques, Theory, and Practice.* ACM SIGGRAPH 2000 Course Notes, 2000

[Bow 04] BOWMAN, Doug A.; KRUIJFF, E.; et al.: *3D User Interfaces. Theory and Practice.* Addison-Wesley, 2004

[Car 97] CAREY, R.; BELL, G.: *The Annotated VRML 2.0 Reference Manual.* Addison-Wesley, 1997

[CAVE] *The CAVE™ at NCSA:* http://cave.ncsa.uiuc.edu/
letzter Abruf: 14.10.2004

[Fav 02] FAVALORA, G. E.; NAPOLI, J.; et al.: *100 Million-voxel volumetric display.* In: Cockpit Displays IX: Displays for Defense Applications. Proceedings of SPIE Vol. 4712, 2002, S. 300-312

[FDT] FIFTH DIMENSION TECHNOLOGIES: *5DT Data Glove 5 Benutzerhandbuch.*
http://www.5dt.com/downloads/5DTDataGlove5Manual.pdf
letzter Abruf: 07.09.2004

[FIRST] FRAUNHOFER FIRST, Internetauftritt: http://www.first.fraunhofer.de/
letzter Abruf: 01.10.2004

[Gib 94] GIBSON, W.: *Neuromancer.* München: Heyne, 1994

[GLV] GLVIEW, Homepage: http://home.snafu.de/hg/
letzter Abruf: 05.08.2004

[Gold 97] GOLDSTEIN, E. Bruce: *Wahrnehmungspsychologie. Eine Einführung.* Heidelberg: Spektrum, Akad. Verl., 1997

[Hart 96] HARMAN, J.; WERNECKE, J.: *The VRML 2.0 Handbook. Building Moving Worlds on the Web.* Addison-Wesley, 1996

[Hux 53] HUXLEY, A.: *Schöne neue Welt.* Frankfurt am Main: Fischer Taschenbuch Verlag, erste Auflage 1953

[INFI] INFITEC GMBH: *Infitec Wellenlängenmultiplex Visualisierungssysteme.* http://www.infitec.net/INFITEC.pdf
letzter Abruf: 11.10.2003

[Isa 02] ISAKOVIC, K.; DUDZIAK, T.; et al.: *X-Rooms. A PC-based immersive visualization environment.* In: Proceeding of the seventh international conference on 3D Web technology, 2002, S. 173-177

[McAll 93] MCALLISTER, David F.: *Stereo computer graphics and other true 3D technologies.* Princeton, New Jersey: Princeton University Press, 1993

[Pau 95] PAUSCH R.; BURNETTEA T.; et al.: *Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures.* In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, S. 399-400

[POL] POLHEMUS: *Polhemus Isotrak II, Fastrak, Patriot Benutzerhandbücher* http://www.polhemus.com/technical_product_support.htm
letzter Abruf: 07.09.2004

[Pou 96] POUPYREV, I.; BILLINGHURST, M.; et al.: *The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR*. In: Proceedings of the 9th annual ACM symposium on User interface software and technology, 1996, S. 79-80

[VRML 97] *Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML)* (ISO/IEC 14772)
http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/
letzter Abruf: 21.09.2004

[Woo 99] WOO, M.; NEIDER, J.; et al.: *OpenGL Programming Guide. The official guide to learning OpenGL, Version 1.2.* Addison-Wesley, 1999

Glossar

CAVE	Cave Automatic Virtual Environment. Ein Raum, auf dessen Wände dreidimensionale Darstellungen projiziert werden. Erlaubt mehreren Benutzern gleichzeitig in eine virtuelle Welt einzutauchen.
DOF	Degree of Freedom. Beschreibt Anzahl von Freiheitsgraden in der VR.
DRS	Distributed Rendering Software. Bei FIRST entwickelte 3D-Cluster-Software basierend auf VRML.
FIRST	Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik.
HMD	Head Mounted Display. Auch als Bildschirmbrille bezeichnet.
IOA	Input Output API. Für das DRS entwickelte Schnittstelle zum Einbinden von Ein-/Ausgabegeräten jeglicher Art.
ISO	International Organization for Standardization. Internationale Vereinigung der Standardisierungsgremien von 148 Ländern.
MFC	Microsoft Foundation Classes. Objektorientierte Bibliothek zum Programmieren von Benutzeroberflächen unter Windows.
MVC	Model-View-Controller. Entwurfsmuster aus der Softwareentwicklung.
PDA	Personal Digital Assistent. Auch Palmtop oder Handheld genannt. Kleiner Taschencomputer.
TCP/IP	Transmission Control Protocol/ Internet Protokoll Familie. De facto Protokollstandard des Internets.
UDP	User Datagram Protocol. Schnelles Protokoll ohne Fehlerkorrektur. Setzt auf TCP/IP auf.

UML	Unified Modeling Language.
VR	Virtuelle Realität. Techniken und Methoden, die es einem Benutzer erlauben in eine computergenerierte Umgebung einzutauchen.
VRML	Virtual Reality Modeling Language. Von der ISO 1997 standardisiertes Dateiformat zum Austausch von 3D-Daten.
Wand	Engl. *Wand* = Zauberstab. In der Hand gehaltenes Eingabegerät für VR-Anwendungen. Besitzt unterschiedliche Anzahl von Knöpfen. Mit Tracking kombiniert.

Abbildungsverzeichnis

Tabellenverzeichnis

A Klassendiagramme der erstellten Anwendungen

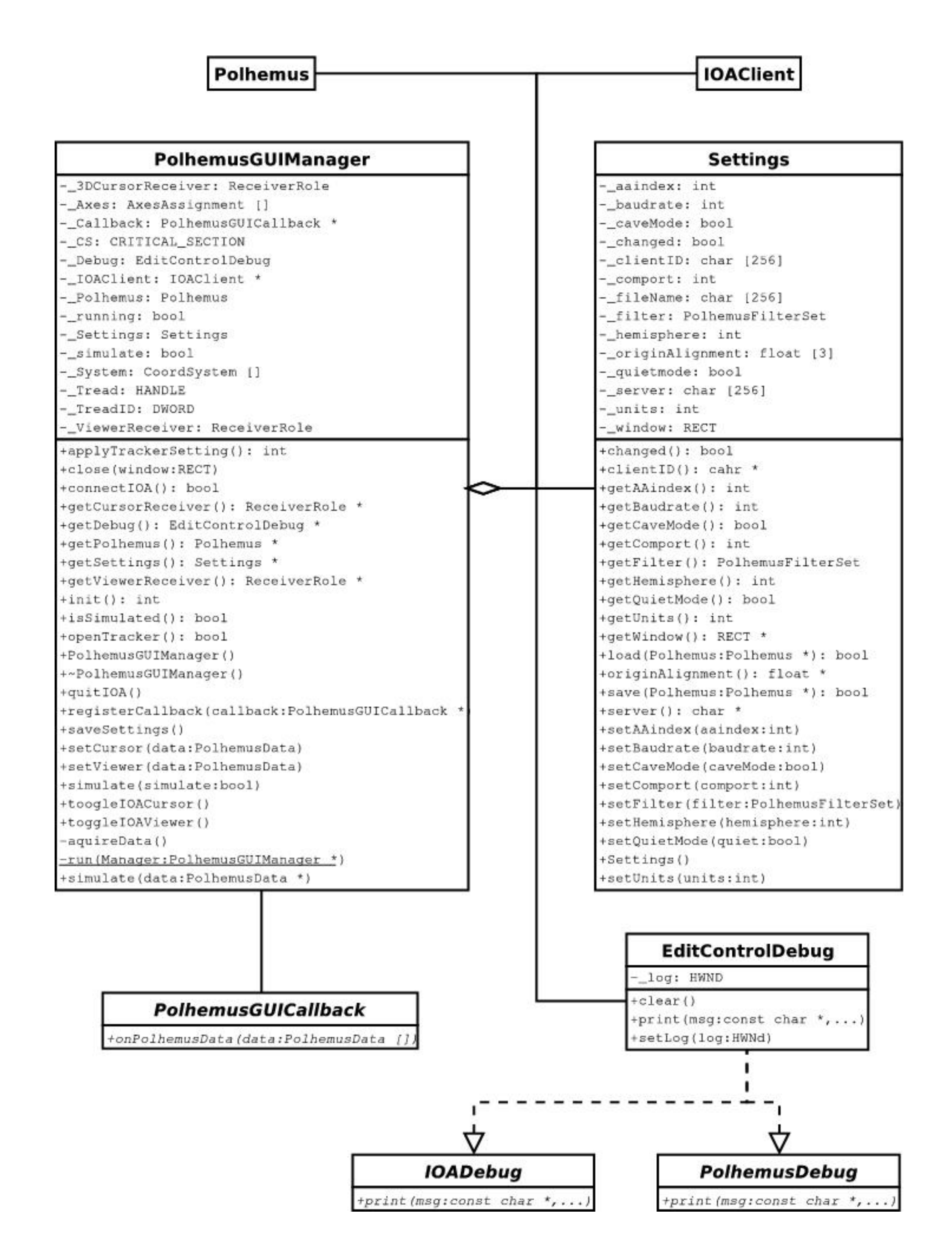


Abbildung A-1: Klassendiagramm PolhemusGUI Ausschnitt 1

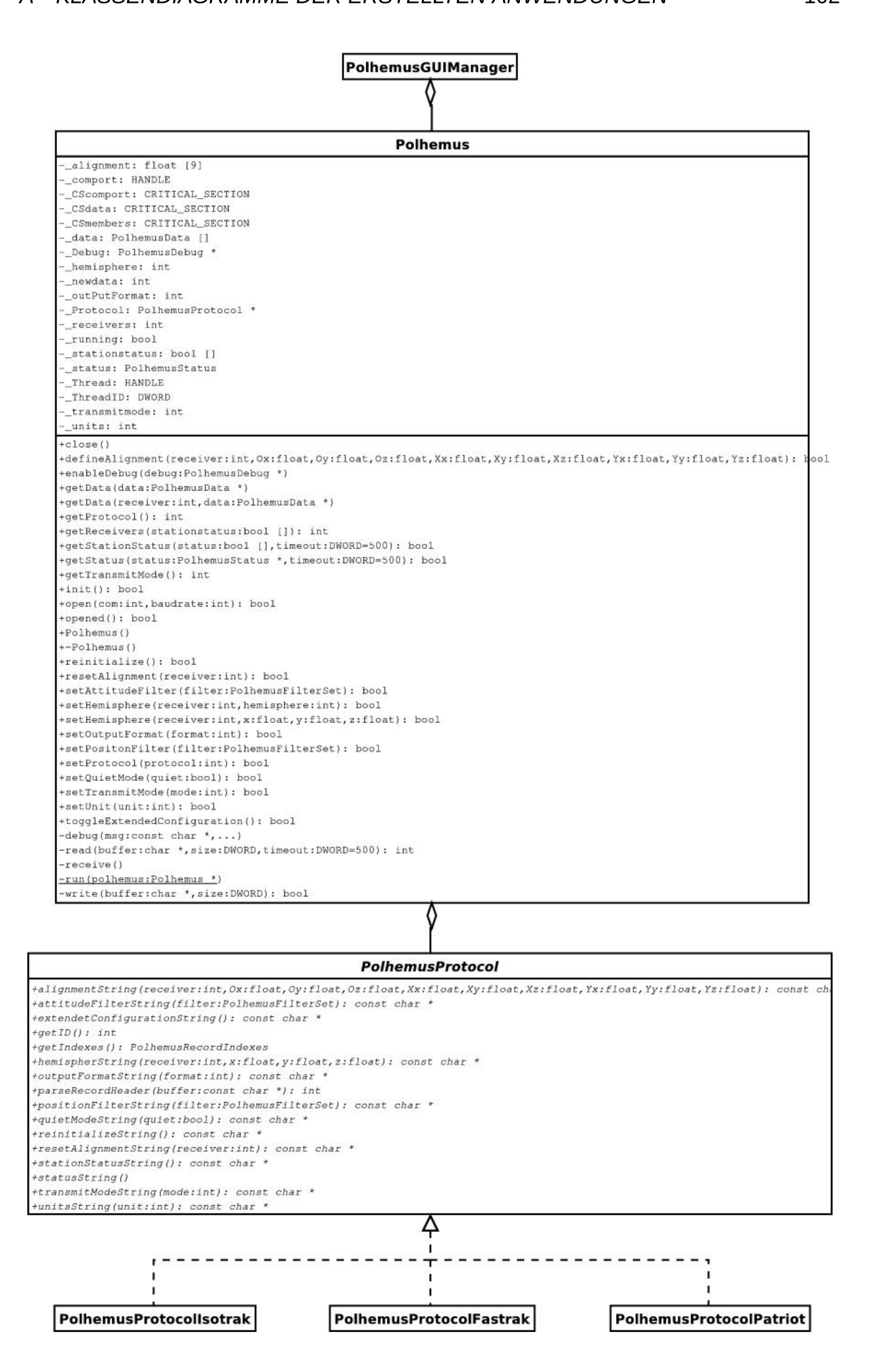


Abbildung A-2: Klassendiagramm PolhemusGUI Ausschnitt 2

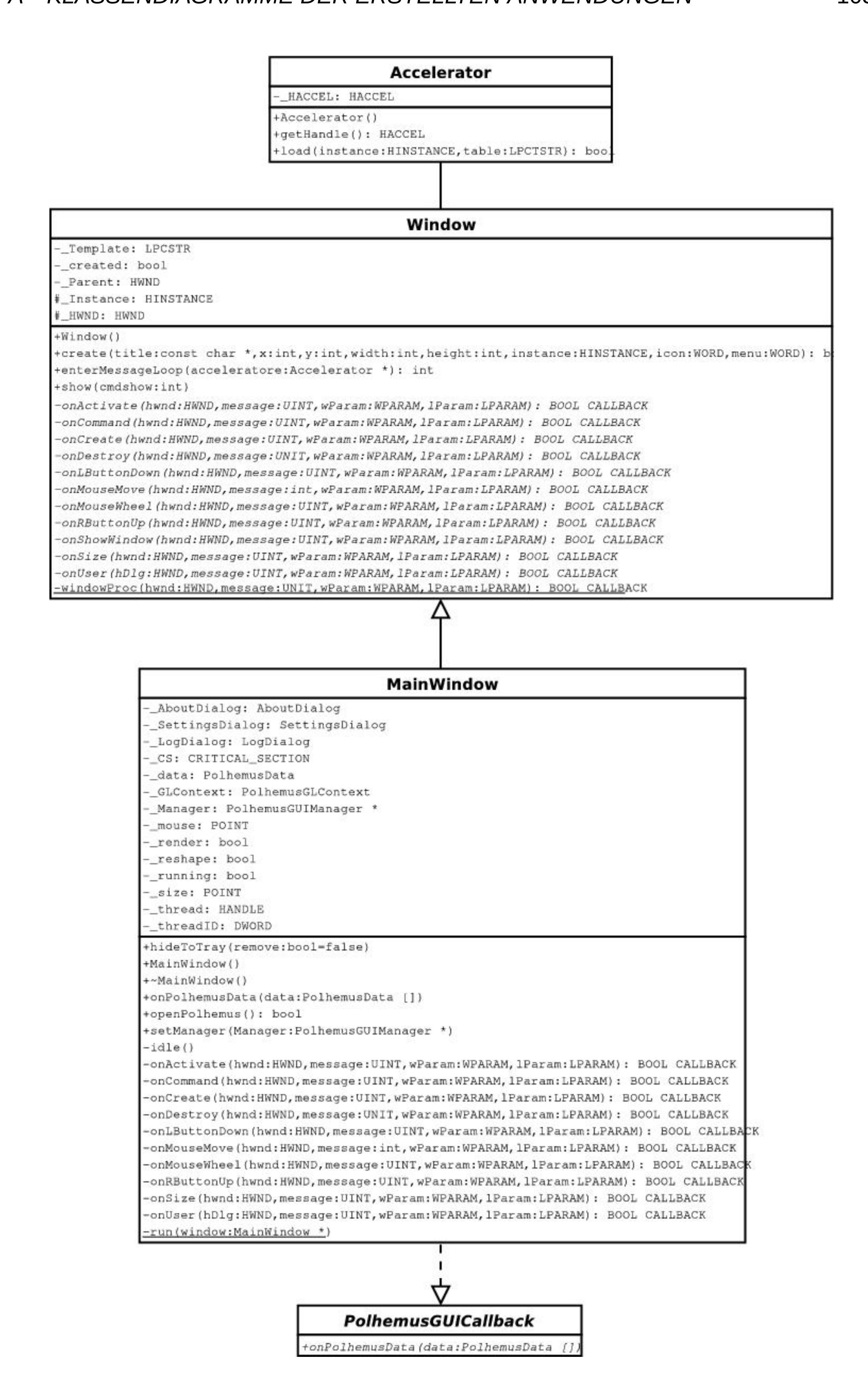


Abbildung A-3: Klassendiagramm PolhemusGUI Ausschnitt 3

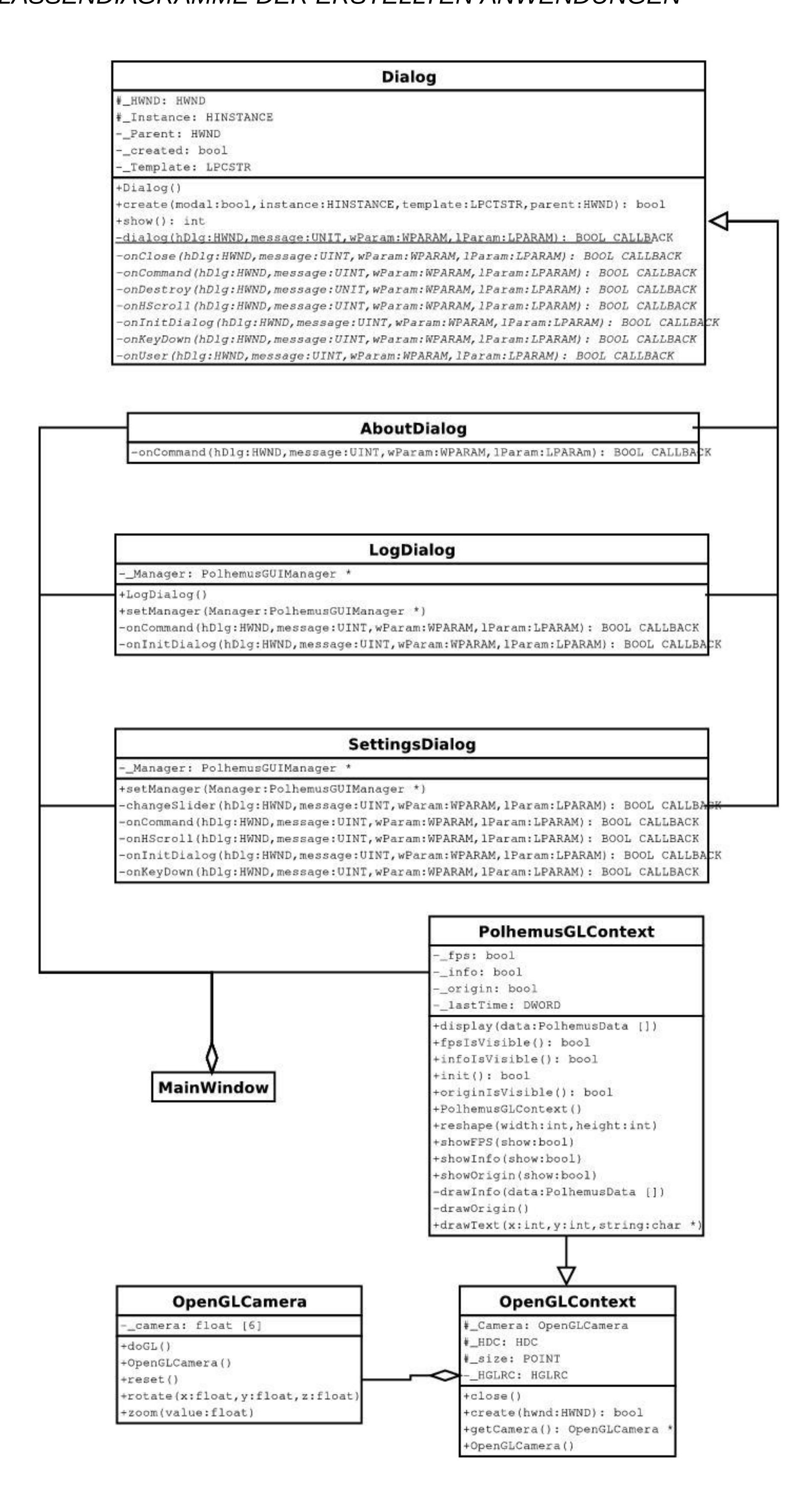


Abbildung A-4: Klassendiagramm PolhemusGUI Ausschnitt 4

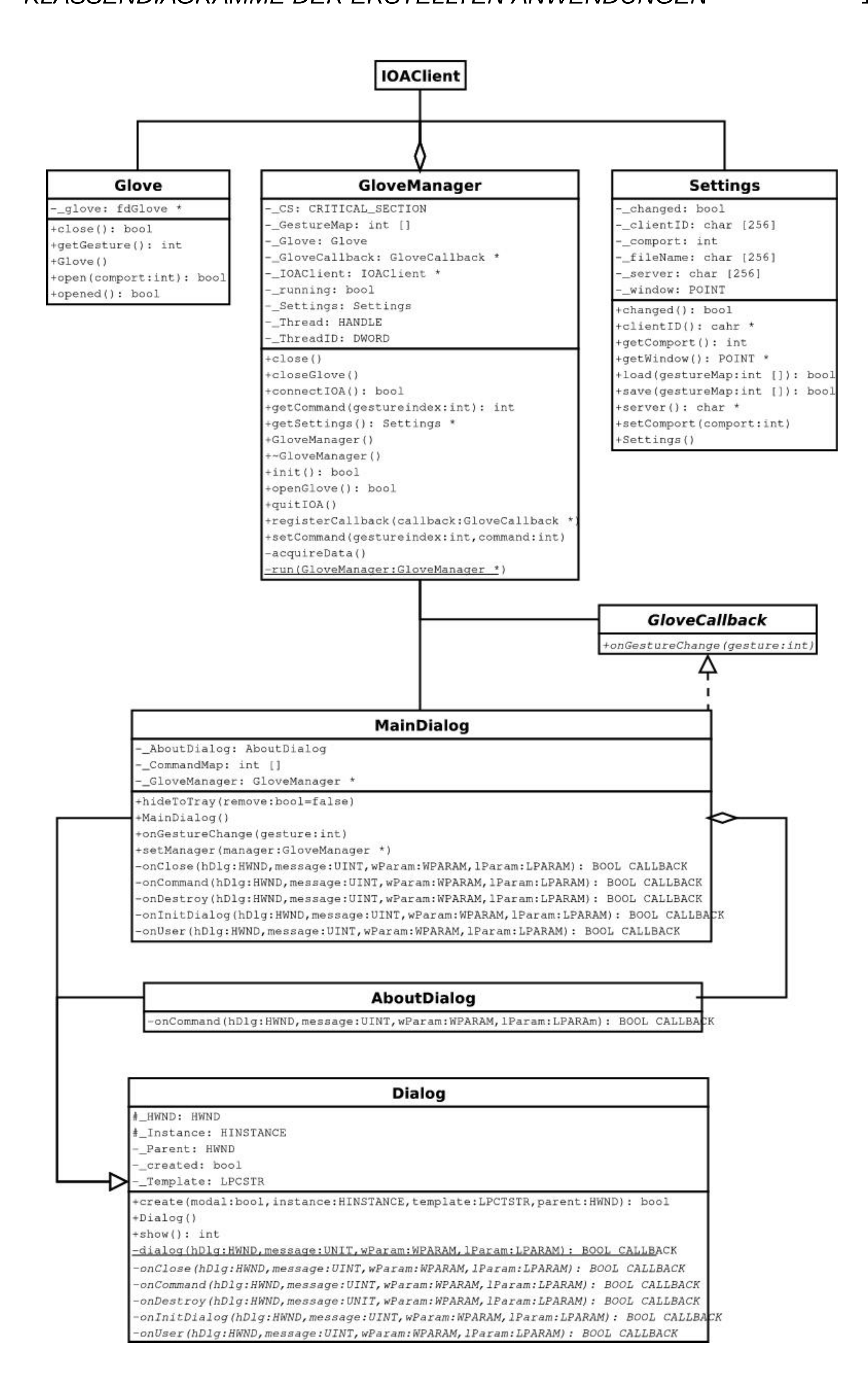


Abbildung A-5: Klassendiagramm Glove vollständig

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Stefan Grüttner

Berlin, 18. November 2004